



Multi Objective Controller Design for Linear Systems via Optimal Interpolation

Hitay Ozbay
Department of Electrical Engineering

NASA Lewis Research Center
Cleveland, Ohio 44135

Grant No. NAG3-1723
Final Progress Report
For the period March 29, 1995 - March 28, 1996
RF Project No. 862718/730689

October, 1996

**This material is based upon work supported by the NASA under
Award No. NAG3-1723.**

MULTI-OBJECTIVE CONTROLLER DESIGN FOR LINEAR SYSTEMS VIA OPTIMAL INTERPOLATION

SUMMARY

We propose a methodology for the design of a controller which satisfies a set of closed-loop objectives simultaneously. The set of objectives consists of (i) pole placement, (ii) decoupled command tracking of step inputs at steady-state, and (iii) minimization of step response transients with respect to envelope specifications. We first obtain a characterization of all controllers placing the closed-loop poles in a prescribed region of the complex plane. In this characterization, the free parameter matrix $Q(s)$ is to be determined to attain objectives (ii) and (iii). Objective (ii) is expressed as determining a Pareto optimal solution to a vector valued optimization problem. The solution of this problem is obtained by transforming it to a scalar convex optimization problem. This solution determines $Q(0)$ and the remaining freedom in choosing $Q(s)$ is used to satisfy objective (iii). We write $Q(s) = (1/v(s))\bar{Q}(s)$ for a prescribed polynomial $v(s)$. $\bar{Q}(s)$ is a polynomial matrix which is arbitrary except that $\bar{Q}(0)$ and the order of $\bar{Q}(s)$ are fixed. Obeying these constraints $\bar{Q}(s)$ is now to be "shaped" to minimize the step response characteristics of specific input/output pairs according to the maximum envelope violations. This problem is expressed as a vector valued optimization problem using the concept of Pareto optimality. We then investigate a scalar optimization problem associated with this vector valued problem and show that it is convex.

The organization of the report is as follows. The next section includes some definitions and preliminary lemmas. We then give the problem statement which is followed by a section including a detailed development of the design procedure. We then consider an aircraft control example. The last section gives some concluding remarks. The Appendix includes the proofs of technical lemmas, printouts of computer programs, and figures.

PRELIMINARIES

We first give some definitions: $\mathcal{R}$ denotes the set of real numbers, $\mathcal{R}_+$ denotes the set of nonnegative real numbers, and $\mathcal{H}$ denotes the set of proper rational functions with real coefficients. The transpose of a matrix E is denoted by E' . If E is a $m \times n$ matrix with entries over a set R , we sometimes denote this by $E \in R^{m \times n}$ or simply by $E \in R$, when the size of E is irrelevant or clear from the context. For a matrix E over $\mathcal{R}$, $\|E\|$ denotes the euclidean norm of E , i.e., $\|E\| = \sqrt{\text{trace}(E'E)}$. I and 0 denote the identity and zero matrices, respectively. For a given set R and matrices $A = [a_{ij}] \in R^{m \times n}$ and $B = [b_{ij}] \in R^{p \times r}$, the product $A \otimes B$ is called the *Kronecker matrix product* and is defined as the following $mp \times nr$ matrix:

$$\begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}.$$

For a transfer matrix $G(s)$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \xleftrightarrow{\text{minimal}} G(s)$$

denotes a minimal state-space realization of $G(s)$ represented by the dynamical equations:

$$\dot{x} = Ax + Bu, \quad y = Cx + Du. \quad (1)$$

Conversely, for a dynamical system as in (1),

$$G(s) \longleftrightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

states that the transfer matrix $G(s)$ satisfies $G(s) = C(sI - A)^{-1}B + D$. For a_1, a_2 positive integers satisfying $a_2 > a_1$, ' $a_1 : a_2$ ' denotes the ordered set of integers ' $\{a_1, \dots, a_2\}$ '. Let v_1 and v_2 be some ordered sets of integers contained in $\{1, \dots, m\}$ and $\{1, \dots, n\}$, respectively. Consider $A \in R^{m \times n}$ for some set R . The notation $A_{(v_1, v_2)}$ defines the submatrix of A containing its rows and columns with indices contained in v_1 and v_2 , respectively. The notation $A_{(:, v_2)}$ (resp. $A_{(v_1, :)}$) defines the subset of A containing its rows and columns with indices in v_1 (resp. $\dots, m$) and $\{1, \dots, n\}$ (resp. v_2). A function $\phi : \mathcal{R}^{m \times n} \rightarrow \mathcal{R}$ is called *convex* if for any $\lambda \in [0, 1]$ and $\alpha_i \in \mathcal{R}^{m \times n}$, $i = 1, 2$ the following inequality holds

$$\phi(\lambda\alpha_1 + (1 - \lambda)\alpha_2) \leq \lambda\phi(\alpha_1) + (1 - \lambda)\phi(\alpha_2)$$



Research Foundation

1960 Kenny Road
Columbus, OH 43210-1063

FINAL TECHNICAL REPORT

to

NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

for

**Multi Objective Controller Design for
Linear Systems via Optimal Interpolation
Grant No. NAG3-1723
March 29, 1995 - March 28, 1996
OSURF RF: 730689**

from

Hitay Ozbay
Department of Electrical Engineering
The Ohio State University
2015 Neil Avenue
Columbus, Ohio 43210

October 30, 1996

For $i = 1, \dots, l$, consider the (cost) function $\phi(K)_i : \mathcal{R}^{m \times n} \rightarrow \mathcal{R}$. We say that $K^* \in \mathcal{R}^{m \times n}$ is *Pareto optimal* with respect to the criterion

$$(\phi(K)_1, \dots, \phi(K)_l; K \in \mathcal{R}^{m \times n})$$

if there does not exist $K^o \in \mathcal{R}^{m \times n}$ satisfying

$$\phi(K^o)_1 \leq \phi(K^*)_1, \dots, \phi(K^o)_l \leq \phi(K^*)_l$$

where strict inequality holds for at least one $i \in \{1, \dots, l\}$.

We will now give three lemmas concerning the concept of Pareto optimality. Lemma 1 states that if the cost functions are the euclidean norms of some vector valued functions the Pareto optimality is preserved when these cost functions are replaced by their second powers. Lemma 2 is concerned with obtaining Pareto optimal solutions via scalarization. Lemma 3 states that the notion of Pareto optimality is preserved under one-to-one and onto mappings.

Lemma 1 Consider the functions $f(K)_i : \mathcal{R}^{m \times n} \rightarrow \mathcal{R}^{k \times l}$, $i = 1, \dots, h$. K^* is Pareto optimal with respect to

$$(\|f(K)_1\|, \dots, \|f(K)_h\|; K \in \mathcal{R}^{m \times n}) \quad (2)$$

if and only if it is Pareto optimal with respect to

$$(\|f(K)_1\|^2, \dots, \|f(K)_h\|^2; K \in \mathcal{R}^{m \times n}). \quad (3)$$

Lemma 2 Consider the functions $\phi(K)_i : \mathcal{R}^{m \times n} \rightarrow \mathcal{R}_+$, $i = 1, \dots, h$, and arbitrary nonnegative numbers α_i , $i = 1, \dots, h$. Any solution K^* of

$$\min_{K \in \mathcal{R}^{m \times n}} \sum_{i=1}^h \alpha_i \phi(K)_i \quad (4)$$

is Pareto optimal with respect to

$$(\phi(K)_1, \dots, \phi(K)_h; K \in \mathcal{R}^{m \times n}). \quad (5)$$

Lemma 3 Consider two sets $\mathcal{X}$ and $\mathcal{Y}$ and assume that there exists a one-to-one function g from $\mathcal{X}$ onto $\mathcal{Y}$. Consider a set of given functions $\phi(K)_i : \mathcal{X} \rightarrow \mathcal{R}$, $i = 1, \dots, h$ and define

$\hat{\phi}(\alpha)_i : \mathcal{Y} \rightarrow \mathcal{R}$, $i = 1, \dots, h$ as $\hat{\phi}(\alpha)_i = \phi(g^{-1}(\alpha))_i$ where $g^{-1}(\cdot)$ denotes the inverse function of $g(\cdot)$ from $\mathcal{Y}$ to $\mathcal{X}$. Then, $K^* \in \mathcal{X}$ is Pareto optimal with respect to

$$(\phi(K)_1, \dots, \phi(K)_h; K \in \mathcal{X}) \quad (6)$$

if and only if so is $g(K^*)$ with respect to

$$(\hat{\phi}(\alpha)_1, \dots, \hat{\phi}(\alpha)_h; \alpha \in \mathcal{Y}) \quad (7)$$

We now give three additional lemmas. Lemma 4 expresses the zero-state unit-step response of a scalar transfer function in terms explicitly of various time functions belonging to different components of the transfer function. Lemmas 5 and 6 consider the convexity of two particular cost functions.

Lemma 4 Consider $A(s) \in \mathcal{H}$, $B(s) \in \mathcal{H}^{1 \times m}$, $Q(s) \in \mathcal{H}^{m \times n}$, and $C(s) \in \mathcal{H}^{n \times 1}$. Assume that there exists a polynomial $v(s)$ of order l such that the (i, j) -th entry of $Q(s)$, denoted by $q_{ij}(s)$, can be written as

$$q_{ij}(s) = \frac{1}{v(s)} (\alpha_{i,j,l} s^l + \alpha_{i,j,l-1} s^{l-1} + \dots + \alpha_{i,j,0}), \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (8)$$

Then, there exists $\bar{\alpha} \in \mathcal{R}^{1 \times mn(l+1)}$, a scalar time function $a(t)$, and a set of scalar time functions consisting of $mn(l+1)$ elements, namely

$$\{f_{1,0}(t), \dots, f_{1,l}(t), f_{2,0}(t), \dots, f_{2,l}(t), \dots, f_{mn,0}(t), \dots, f_{mn,l}(t)\},$$

such that the zero-state unit step response associated with the transfer function $A(s) + B(s)Q(s)C(s)$ can be written as

$$a(t) + \bar{\alpha}[f_{1,0}(t) \dots f_{1,l}(t) f_{2,0}(t) \dots f_{2,l}(t) \dots f_{mn,0}(t) \dots f_{mn,l}(t)]'.$$

Lemma 5 Consider scalar time functions $a(t)$, $b_i(t)$, $i = 1, \dots, n$ which are bounded in $t \geq 0$. For $\alpha \in \mathcal{R}^{1 \times n}$ define a cost function $\phi(\alpha)$ as follows:

$$\phi(\alpha) = \sup_{t \geq 0} \{a(t) + \alpha[b_1(t) \dots b_n(t)]'\}.$$

The function $\phi(\alpha)$ is convex in α .

Lemma 6 Consider scalar time functions $a_1(t)$, $a_2(t)$, $b_i(t)$, $c_i(t)$, $i = 1, \dots, n$ which are bounded in $t \geq 0$. For $\alpha \in \mathcal{R}^{1 \times n}$ define the cost function $\phi(\alpha)$ as follows:

$$\phi(\alpha) = \sup_{t \geq 0} \max\{a_1(t) + \alpha[b_1(t) \dots b_n(t)]', a_2(t) + \alpha[c_1(t) \dots c_n(t)]', 0\}.$$

The function $\phi(\alpha)$ is convex in α .

The proofs of the lemmas can be found in Appendix A.

PROBLEM STATEMENT

Consider the feedback system in Figure 1 where w_1 , w_2 are the disturbance inputs and z_1 , z_2 are the regulated outputs. In general, we want to minimize the effect of the disturbances on the regulated outputs in the closed-loop system. The output y is the measured output and u is the control input. The plant $G(s)$ and the controller $K(s)$ are linear time-invariant finite dimensional systems. The transfer matrix associated with the input/output pair (u, y) is strictly proper. Let p_1 , p_2 , r_1 , and r_2 denote the dimensions of the vectors z_1 , z_2 , w_1 , and w_2 , respectively. For simplicity, we will be concerned with only those plants satisfying $p_1 = r_1 = r_2 = 1$, $p_2 = 2$. Our discussion can be extended to more general classes of systems in a straightforward way.

Let a subset $\mathcal{P}$ of the left half complex plane be given. This set prescribes the desired closed-loop pole locations. $\mathcal{P}$ is arbitrary except that it satisfies several assumptions made for technical reasons. First, $\mathcal{P}$ is symmetric, i.e., if a is contained in $\mathcal{P}$ then so is the complex conjugate of a . Secondly, we assume that the unobservable and/or uncontrollable modes of $G(s)$ around the control channel (u, y) are all contained in $\mathcal{P}$. Finally, the number of elements of $\mathcal{P}$ is no less than the order (total number of poles with multiplicities) of the open-loop plant $G(s)$ and $\mathcal{P}$ contains at least one real element.

Let $\mathcal{K}$ denote the set of all controllers which satisfy that the closed-loop poles are contained in $\mathcal{P}$.

Some arbitrary time functions

$$s_1^M(t), s_1^m(t), s_{21}^M(t), s_{21}^m(t), s_{22}^M(t), s_{22}^m(t) \quad (9)$$

are given. These functions are defined for $t \geq 0$ and are continuous in their domain of definition. It is assumed that they satisfy

$$s_1^M(t) \geq s_1^m(t), s_{21}^M(t) \geq s_{21}^m(t), s_{22}^M(t) \geq s_{22}^m(t), \forall t \geq 0$$

and

$$\lim_{t \rightarrow \infty} s_1^M(t) = \lim_{t \rightarrow \infty} s_1^m(t) = \lim_{t \rightarrow \infty} s_{21}^M(t) = \lim_{t \rightarrow \infty} s_{21}^m(t) = \lim_{t \rightarrow \infty} s_{22}^M(t) = \lim_{t \rightarrow \infty} s_{22}^m(t) = 0$$

The functions (9) are called the *envelope* functions. (See below.)

We seek for a controller $K^*(s)$ which satisfies the following set of design objectives:

I. Pole placement: We require

$$K^*(s) \in \mathcal{K}.$$

II. Decoupled command tracking of step inputs at steady-state: Let $C_1[K(s)]$ and $C_2[K(s)]$ denote the closed-loop transfer functions associated with the input/output pairs (w_1, z_1) and (w_2, z_2) , respectively, and define

$$e_1(K) = \lim_{s \rightarrow 0} C_1[K(s)], \quad e_2(K) = \lim_{s \rightarrow 0} C_2[K(s)].$$

The controller $K^*(s)$ should be Pareto optimal with respect to the criterion

$$(\|e_1(K(s))\|, \|e_2(K(s))\|; K(s) \in \mathcal{K}).$$

III. Minimization of step response transients with respect to envelope specifications: Let z_2 be partitioned as $z_2 = [z_{21} \ z_{22}]'$. Consider a fixed but otherwise arbitrary $K \in \mathcal{K}$. Assume that the closed-loop system is initially at rest and apply the following input:

$$w_1(t) = 1, \ w_2(t) = 0, \ \forall t > 0.$$

The corresponding response of z_1 is denoted by

$$c_1(t) = z_1(t).$$

Now, again assume that the closed-loop system is initially at rest and apply the following input:

$$w_1(t) = 0, \ w_2(t) = 1, \ \forall t > 0.$$

Denote the corresponding response of $z_2(t)$ by

$$c_{21}(t) = z_{21}(t), \ c_{22}(t) = z_{22}(t).$$

Note that the functions $c_1(t)$, $c_{21}(t)$ and $c_{22}(t)$ are bounded in $[0, \infty)$, due to the fact that they are the step responses of stable proper transfer functions. Also notice that $c_1(t)$, $c_{21}(t)$ and $c_{22}(t)$ are functions of $K(s)$; we suppress this for notational simplicity.

We define the *maximum envelope violations* as follows:

$$\phi(K(s))_1 = \sup_{t \geq 0} \max\{c_1(t) - s_1^M(t), s_1^m(t) - c_1(t), 0\}, \quad (10)$$

$$\phi(K(s))_{21} = \sup_{t \geq 0} \max\{c_{21}(t) - s_{21}^M(t), s_{21}^m(t) - c_{21}(t), 0\}, \quad (11)$$

$$\phi(K(s))_{22} = \sup_{t \geq 0} \max\{c_{22}(t) - s_{22}^M(t), s_{22}^m(t) - c_{22}(t), 0\}. \quad (12)$$

The cost function in (10) equals the maximum of the deviation of the closed-loop step response associated with (w_1, z_1) from the region upper and lower bounded by the corresponding envelope functions. Similar interpretations can be given for the cost functions in (11) and (12).

Let a subset $\hat{\mathcal{K}}$ of $\mathcal{K}$ be given. This set is *not* arbitrary. We will give a more precise description of $\hat{\mathcal{K}}$ in the next section.

Our third objective can now be stated.

We require $K^*(s) \in \hat{\mathcal{K}}$ and that $K^*(s)$ is Pareto optimal with respect to the criterion

$$(\phi(K(s))_1, \phi(K(s))_{21}, \phi(K(s))_{22}; K(s) \in \hat{\mathcal{K}}). \quad (13)$$

AN OPTIMAL INTERPOLATION APPROACH

Let us first consider how to achieve the objectives I and II.

Let

$$\left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array} \right] \xleftrightarrow{\text{minimal}} G(s)$$

such that B_1 and B_2 have r_1 and r_2 columns, respectively, and C_1 and C_2 have p_1 and p_2 rows, respectively. Let F and H be matrices of suitable dimensions such that the eigenvalues of $A_F := A + B_2 F$ and $A_H := A + H C_2$ are contained in $\mathcal{P}$. Let

$$\bar{A} := \begin{bmatrix} A_F & -B_2 F \\ 0 & A_H \end{bmatrix}, \quad \bar{B} := \begin{bmatrix} B_1 \\ B_1 + H D_{21} \end{bmatrix}, \quad \bar{C} := \begin{bmatrix} C_1 + D_{12} F & -D_{12} F \end{bmatrix}. \quad (14)$$

Define the following transfer matrices

$$R_1(s) \longleftarrow \begin{bmatrix} \bar{A} & \bar{B}_{(:,1:r_1)} \\ \bar{C}_{(1:p_1,:)} & D_{11(1:p_1,1:r_1)} \end{bmatrix}, \quad R_2(s) \longleftarrow \begin{bmatrix} \bar{A} & \bar{B}_{(:,r_1+1:r_1+r_2)} \\ \bar{C}_{(p_1+1:p_1+p_2,:)} & D_{11(p_1+1:p_1+p_2,r_1+1:r_1+r_2)} \end{bmatrix}, \quad (15)$$

$$U_1(s) \longleftrightarrow \begin{bmatrix} A_F & B_2 \\ C_{1(1:p_1,:)} + D_{12(1:p_1,:)}F & D_{12(1:p_1,:)} \end{bmatrix}, \quad (16)$$

$$U_2(s) \longleftrightarrow \begin{bmatrix} A_F & B_2 \\ C_{1(p_1+1:p_1+p_2,:)} + D_{12(p_1+1:p_1+p_2,:)}F & D_{12(p_1+1:p_1+p_2,:)} \end{bmatrix}, \quad (17)$$

$$V_1(s) \longleftrightarrow \begin{bmatrix} A_H & B_{1(:,1:r_1)} + HD_{21(:,1:r_1)} \\ C_2 & D_{21(:,1:r_1)} \end{bmatrix}, \quad V_2(s) \longleftrightarrow \begin{bmatrix} A_H & B_{1(:,r_1+1:r_1+r_2)} + HD_{21(:,r_1+1:r_1+r_2)} \\ C_2 & D_{21(:,r_1+1:r_1+r_2)} \end{bmatrix}, \quad (18)$$

$$T_X(s) \longleftrightarrow \begin{bmatrix} A_H & -B_2 \\ F & I \end{bmatrix}, \quad T_Y(s) \longleftrightarrow \begin{bmatrix} A_H & -H \\ F & 0 \end{bmatrix}, \quad (19)$$

$$T_N(s) \longleftrightarrow \begin{bmatrix} A_H & B_2 \\ C_2 & 0 \end{bmatrix}, \quad T_M(s) \longleftrightarrow \begin{bmatrix} A_H & H \\ C_2 & I \end{bmatrix}. \quad (20)$$

A controller $K(s)$ is an element of $\mathcal{K}$ if and only if

$$K(s) = (T_X(s) - Q(s)T_N(s))^{-1}(T_Y(s) - Q(s)T_M(s)) \quad (21)$$

for some proper rational matrix $Q(s)$ whose poles are contained in $\mathcal{P}$. Moreover, the transfer function associated with (z_1, w_1) is equal to $R_1(s) + U_1(s)Q(s)V_1(s)$ and the transfer matrix associated with (z_2, w_2) is equal to $R_2(s) + U_2(s)Q(s)V_2(s)$.

Objective II can now be more explicitly stated as follows: Determine a proper rational matrix $Q(s)$ whose poles are contained in $\mathcal{P}$ and $Q(0)$ is Pareto optimal with respect to the criterion

$$(\|R_1(0) + U_1(0)Q(0)V_1(0)\|, \|R_2(0) + U_2(0)Q(0)V_2(0)\|; Q(0) \in \mathcal{R}) \quad (22)$$

We attack the problem of determining a Pareto optimal $Q(0)$ by transforming that problem to a scalar minimization problem. Note from Lemma 1 that $Q(0)$ is Pareto optimal with respect to the criterion (22) if and only if it is Pareto optimal with respect to the criterion

$$(\|R_1(0) + U_1(0)Q(0)V_1(0)\|^2, \|R_2(0) + U_2(0)Q(0)V_2(0)\|^2; Q(0) \in \mathcal{R}) \quad (23)$$

We solve

$$\min_{S \in \mathcal{R}} \alpha_1 \|R_1(0) + U_1(0)SV_1(0)\|^2 + \alpha_2 \|R_2(0) + U_2(0)SV_2(0)\|^2 \quad (24)$$

for any nonnegative α_1 and α_2 satisfying $\alpha_1 + \alpha_2 = 1$, and let $Q(0) = S^*$ where S^* is the solution of the minimization problem in (24). Defined this way, $Q(0)$ is a Pareto optimal solution with respect to the criterion (23) via Lemma 2.

The reason for dealing with problem (23) rather than (22) is purely technical. The removal of the square-root operation speeds up the computation.

The cost function to be minimized in (24) is convex in S . The solution of (24) can be obtained via the `fminu` function of MATLAB. The constants α_1 and α_2 in (24) let us give different weightings to the cost functions $\|R_1(0) + U_1(0)Q(0)V_1(0)\|$ and $\|R_2(0) + U_2(0)Q(0)V_2(0)\|$.

We arrive at the following conclusion:

A controller $K(s)$ satisfies objective II (together with objective I) if (21) is satisfied for some proper rational matrix $Q(s)$ such that the poles of $Q(s)$ are contained in $\mathcal{P}$ and $Q(0) = S^$.*

It is seen that there is a considerable freedom in designing a controller $K(s)$ to satisfy objectives I and II. We will now deal with objective III to make use of this freedom.

Let us define a subset $\hat{\mathcal{K}}$ of $\mathcal{K}$ as promised in the previous section. Let $v(s)$ be a polynomial of order l such that the roots of $v(s)$ are all contained in $\mathcal{P}$. Define

$$\hat{\mathcal{K}} = \{K(s) \mid K(s) = (T_X(s) - Q(s)T_N(s))^{-1}(T_Y(s) - Q(s)T_M(s)), Q(s) \text{ is proper rational, } Q(s) = (1/v(s))\bar{Q}(s) \text{ for some polynomial matrix } \bar{Q}(s), \text{ and } Q(0) = S^*\}$$

The (i, j) -th entry $q_{ij}(s)$ of $Q(s)$ can be written as in (8). Note that

$$Q(0) = \frac{1}{v(s)} \begin{bmatrix} \alpha_{1,1,0} & \dots & \alpha_{1,n,0} \\ \vdots & & \vdots \\ \alpha_{m,1,0} & \dots & \alpha_{m,n,0} \end{bmatrix}$$

A bijective map between $\hat{\mathcal{K}}$ and $\mathcal{R}^{1 \times mn l}$.

We will now show the existence of a one-to-one function from $\hat{\mathcal{K}}$ onto $\mathcal{R}^{1 \times mn l}$. This function will be used to find a Pareto optimal solution with respect to the criteria (13).

From $\hat{\mathcal{K}}$ to $\mathcal{R}^{1 \times mn l}$:

(a1) For $K(s) \in \hat{\mathcal{K}}$, define

$$Q(s) = (T_X(s)K(s) - T_Y(s))(T_N(s)K(s) - T_M(s))^{-1} \quad (25)$$

Each entry of $Q(s)$ is in the form of (8) where $\alpha_{i,j,0}$, $i = 1, \dots, m$, $j = 1, \dots, n$, satisfy

$$\begin{bmatrix} \alpha_{1,1,0} & \dots & \alpha_{1,n,0} \\ \vdots & & \vdots \\ \alpha_{m,1,0} & \dots & \alpha_{m,n,0} \end{bmatrix} = v(0)S^* \quad (26)$$

(a2) Define $\bar{\alpha}_{i,j}$, $i = 1, \dots, mn$, $j = 1, \dots, l$ as in (38).

(a3) Define $\alpha \in \mathcal{R}^{1 \times mn l}$ as

$$\alpha = [\bar{\alpha}_{1,1} \dots \bar{\alpha}_{1,l} \bar{\alpha}_{2,1} \dots \bar{\alpha}_{2,l} \dots \bar{\alpha}_{mn,1} \dots \bar{\alpha}_{mn,l}]. \quad (27)$$

From $\mathcal{R}^{1 \times mn l}$ to $\hat{\mathcal{K}}$:

(b1) Represent $\alpha \in \mathcal{R}^{1 \times mn l}$ as in (27).

(b2) Obtain $\alpha_{i,j,0}$, $i = 1, \dots, m$, $j = 1, \dots, n$ to satisfy (26).

(b3) Define $\alpha_{i,j,k}$, $i = 1, \dots, m$, $j = 1, \dots, n$, $k = 1, \dots, l$ as in (38) in terms of $\bar{\alpha}_{i,j}$, $i = 1, \dots, mn$, $j = 1, \dots, l$.

(b4) Construct $Q(s) = [q_{ij}(s)]$ via (8). $Q(s)$ is proper rational, $Q(s) = (1/v(s))\bar{Q}(s)$ for some polynomial matrix $\bar{Q}(s)$, and $Q(0) = S^*$.

(b5) Define $K(s)$ as in (21), which is contained in $\hat{\mathcal{K}}$.

Let $g(\cdot)$ be the map from $\hat{\mathcal{K}}$ to $\alpha \in \mathcal{R}^{1 \times mn l}$ defined via (a1)-(a3). It is not difficult to show that $g(\cdot)$ is one-to-one and onto. The inverse map $g^{-1}(\cdot)$ of $g(\cdot)$ is given by (b1)-(b5).

Let us now obtain equivalent representations of the cost functions in (10)-(12) in terms of a parameter over $\mathcal{R}^{1 \times mn l}$.

Partition $R_2(s)$ and $U_2(s)$ as $R_2(s) = [R_{21}(s) \ R_{22}(s)]'$ and $U_2(s) = [U'_{21}(s) \ U'_{22}(s)]'$. Then, $c_1(t)$, $c_{21}(t)$, and $c_{22}(t)$ are the zero-state unit-step responses associated with the transfer functions $R_1(s) + U_1(s)Q(s)V_1(s)$, $R_{21}(s) + U_{21}(s)Q(s)V_2(s)$, and $R_{22}(s) + U_{22}(s)Q(s)V_2(s)$, respectively.

Write the (i, j) -th entry $q_{ij}(s)$ of $Q(s)$ as in (8). Define $\bar{\alpha}$ as in (39). From Lemma 4, it follows that there exists scalar valued time functions $\bar{a}_i(t)$, $i = 1, 2, 3$, and vector valued time functions $\bar{f}_i(t)$, $i = 1, 2, 3$, each of dimension $1 \times mn(l+1)$, such that

$$c_1(t) = \bar{a}_1(t) + \bar{\alpha}\bar{f}_1(t), \ c_{21}(t) = \bar{a}_2(t) + \bar{\alpha}\bar{f}_2(t), \ c_{22}(t) = \bar{a}_3(t) + \bar{\alpha}\bar{f}_3(t), \ t \geq 0 \quad (28)$$

From the constructive proof of Lemma 4, one can easily obtain the exact expressions of the time functions $\bar{a}_i(t)$, and $\bar{f}_i(t)$, $i = 1, 2, 3$. Note that

$$Q(0) = (1/v(0)) \begin{bmatrix} \bar{\alpha}_{1,0} & \dots & \bar{\alpha}_{mn-m+1,0} \\ \vdots & & \vdots \\ \bar{\alpha}_{m,0} & \dots & \bar{\alpha}_{mn,0} \end{bmatrix}$$

That is, the elements $\bar{\alpha}_{i,0}$, $i \in \{1, \dots, mn\}$ are fixed to satisfy $Q(0) = S^*$. Define a vector α in $\mathcal{R}^{1 \times mn}$ according to (27). Observe that α is constructed from $\bar{\alpha}$ by deleting its entries corresponding to $\bar{\alpha}_{i,0}$, $i \in \{1, \dots, mn\}$; α represents the *free* elements of $Q(s)$.

We then rewrite (28) as

$$c_1(t) = a_1(t) + \alpha f_1(t), \quad c_{21}(t) = a_2(t) + \alpha f_2(t), \quad c_{22}(t) = a_3(t) + \alpha f_3(t), \quad t \geq 0 \quad (29)$$

for some scalar time functions $a_i(t)$, $i = 1, 2, 3$ and vector valued time functions $f_i(t)$. The explicit expressions for $a_i(t)$ and $f_i(t)$, $i = 1, 2, 3$ are omitted. They can be obtained from $\bar{a}_i(t)$, and $\bar{f}_i(t)$, $i = 1, 2, 3$ using the definition of α .

We define the following cost functionals mapping $\mathcal{R}^{1 \times mn}$ to $\mathcal{R}_+$:

$$\hat{\phi}(\alpha)_1 = \phi(g^{-1}(\alpha))_1, \quad \hat{\phi}(\alpha)_{21} = \phi(g^{-1}(\alpha))_{21}, \quad \hat{\phi}(\alpha)_{22} = \phi(g^{-1}(\alpha))_{22}.$$

For $K(s) \in \hat{\mathcal{K}}$,

$$\phi(K(s))_1 = \hat{\phi}(g(K(s)))_1, \quad \phi(K(s))_{21} = \hat{\phi}(g(K(s)))_{21}, \quad \phi(K(s))_{22} = \hat{\phi}(g(K(s)))_{22}$$

and, via Lemma 3, a controller $K(s)$ is Pareto optimal with respect to (13) if and only if $g(K(s))$ is Pareto optimal with respect to

$$(\hat{\phi}(\alpha)_1, \hat{\phi}(\alpha)_{21}, \hat{\phi}(\alpha)_{22}; \alpha \in \mathcal{R}^{1 \times mn}) \quad (30)$$

This transforms the problem of finding a Pareto optimal controller with respect to (13) to that of finding a Pareto optimal vector in $\mathcal{R}^{1 \times mn}$ with respect to (30). By Lemma 6, each of the cost functionals $\hat{\phi}(\alpha)_1$, $\hat{\phi}(\alpha)_{21}$, and $\hat{\phi}(\alpha)_{22}$ is convex in α . Since the linear combination of convex functionals is also convex, for any positive numbers $\lambda_1, \lambda_2, \lambda_3$, satisfying $\lambda_1 + \lambda_2 + \lambda_3 = 1$, the cost functional

$$\phi(\alpha) := \lambda_1 \hat{\phi}(\alpha)_1 + \lambda_2 \hat{\phi}(\alpha)_{21} + \lambda_3 \hat{\phi}(\alpha)_{22} \quad (31)$$

is convex in α . The numerical issues concerning how to find α^* minimizing (31) are not considered in this report. Assuming a numerical procedure is available to obtain such an α^* , we summarize the design procedure as follows:

Step 1. Determine the matrices F and H such that the eigenvalues of $A + B_2F$ and $A + HC_2$ are contained in $\mathcal{P}$. Determine $\bar{A}$, $\bar{B}$, $\bar{C}$, $R_1(s)$, $R_2(s)$, $U_1(s)$, $U_2(s)$, $V_1(s)$, $V_2(s)$, $T_X(s)$, $T_Y(s)$, $T_N(s)$, $T_M(s)$ according to (14)-(20).

Step 2. Determine the weighting elements α_1 and α_2 according to the design requirements, e.g., desired trade-offs between the channels and solve the scalar optimization problem (24) for S^* .

Step 3. Obtain the time functions $a_i(t)$, $f_i(t)$, $i = 1, 2, 3$. Determine the weighting elements λ_i , $i = 1, 2, 3$. Using the result of scalar optimization problem (31) obtain a Pareto optimal α with respect to (30). Let α be represented as in (27).

Step 4. Obtain $\alpha_{i,j,0}$, $i = 1, \dots, m$, $j = 1, \dots, n$, from (26) and $\alpha_{i,j,k}$, $i = 1, \dots, m$, $j = 1, \dots, n$, $k = 1, \dots, l$, from (38), in terms of $\bar{\alpha}_{i,j}$, $i = 1, \dots, mn$, $j = 1, \dots, l$.

Step 5. Using $\alpha_{i,j,k}$ construct a stable proper matrix $Q(s) = [q_{ij}(s)]$ according to (8).

Step 6. Finally, let

$$K^*(s) = (T_X(s) - Q(s)T_N(s))^{-1}(T_Y(s) - Q(s)T_M(s)).$$

$K^*(s)$ satisfies objectives I-III simultaneously.

DESIGN EXAMPLE

Consider Figure 2. The system represents the simplified lateral/directional dynamics of a very large four-engined passenger aircraft at a particular operating point at high altitude and high longitudinal speed. (See Chapter 10.6.5 of McLean, Automatic Flight Control Systems, 1990). w_1 and w_2 are the reference inputs for the yaw rate and bank angle commands, respectively. z_1 is the yaw rate error, i.e.,

$$z_1 = w_1 - \text{yaw rate},$$

z_{21} is the bank angle error, i.e.,

$$z_{21} = w_2 - \text{bank angle},$$

and z_{22} is the slip angle. We denote the yaw rate, bank angle, and slip angle by r , ϕ , and γ , respectively. The control input u is composed of the aileron and rudder angles (δ_a and δ_r , respectively) and the measured output vector y is composed of r , ϕ and the roll rate p . ($p = \dot{\phi}$.) The states of the linear system are r , ϕ , p , and γ .

It is desired to design a controller which yields asymptotic yaw rate command tracking and bank angle command tracking for step inputs while the slip angle is isolated from the bank angle command at steady state. That is, letting ε denote a very small number, the transfer matrix from $[w_1 \ w_2]'$ to $[z_1 \ z_{21} \ z_{22}]'$ should be equal to

$$\begin{bmatrix} \varepsilon & \times \\ \times & \varepsilon \\ \times & \varepsilon \end{bmatrix}$$

as $s \rightarrow 0$ where we do not care the entries marked by $\times$. Expressing this design objective in terms of standard H_∞ or H_2 control problems is difficult because only the (block) main diagonal elements of the closed-loop transfer matrix are being minimized.

We design a controller to achieve objectives I and II. We have developed a computer program implementing steps 1 and 2 above (Appendix B). The set of desired closed-loop poles is chosen as

$$\mathcal{P} = \{-0.8, -1.9, -1.5 \pm 2.7j, -1.3, -1, -1.7 \pm 2.7j\}.$$

The controller minimizes the magnitudes of the transfer functions associated with the pairs (w_1, z_1) , (w_2, z_{21}) , and (w_2, z_{22}) at $s_0 = 0$, $s_1 = 2\pi/5j \text{ rad/s} (= 1.256j \text{ rad/s})$, and $s_2 = 2\pi/3j \text{ rad/s} (= 2.093j \text{ rad/s})$.

We are looking for $Q(s_0)$, $Q(s_1)$, and $Q(s_2)$, satisfying that $Q(s_0)$ is Pareto optimal with respect to

$$(\|R_1(s_0) + U_1(s_0)Q(s_0)V_1(s_0)\|, \|R_2(s_0) + U_2(s_0)Q(s_0)V_2(s_0)\|; Q(s_0) \in \mathcal{R}), \quad (32)$$

$Q(s_1)$ is Pareto optimal with respect to

$$(\|R_1(s_1) + U_1(s_1)Q(s_1)V_1(s_1)\|, \|R_2(s_1) + U_2(s_1)Q(s_1)V_2(s_1)\|; Q(s_1) \in \mathcal{R}), \quad (33)$$

and $Q(s_2)$ is Pareto optimal with respect to

$$(\|R_1(s_2) + U_1(s_2)Q(s_2)V_1(s_2)\|, \|R_2(s_2) + U_2(s_2)Q(s_2)V_2(s_2)\|; Q(s_2) \in \mathcal{R}). \quad (34)$$

We define the scalar optimization problems in (35), (36), (37) associated with (32), (33), and (34), respectively:

$$\min_{Q(s_0) \in \mathcal{R}} (\alpha_1 \|R_1(s_0) + U_1(s_0)Q(s_0)V_1(s_0)\| + \alpha_2 \|R_2(s_0) + U_2(s_0)Q(s_0)V_2(s_0)\|), \quad (35)$$

$$\min_{Q(s_1) \in \mathcal{R}} (\alpha_1 \|R_1(s_1) + U_1(s_1)Q(s_1)V_1(s_1)\| + \alpha_2 \|R_2(s_1) + U_2(s_1)Q(s_1)V_2(s_1)\|) \quad (36)$$

$$\min_{Q(s_2) \in \mathcal{R}} (\alpha_1 \|R_1(s_2) + U_1(s_2)Q(s_2)V_1(s_2)\| + \alpha_2 \|R_2(s_2) + U_2(s_2)Q(s_2)V_2(s_2)\|) \quad (37)$$

After obtaining minimizing solutions $Q^*(s_0)$, $Q^*(s_1)$, and $Q^*(s_2)$ to (35), (36), (37), respectively (these are Pareto optimal solutions of (32), (33), (34) via Lemma 2), we determine a stable rational matrix $Q(s)$ which satisfies $Q(s_i) = Q^*(s_i)$, $i = 0, 1, 2$, and obtain the controller $K(s) = (T_X(s) - Q(s)T_N(s))^{-1}(T_Y(s) - Q(s)T_M(s))$.

Figures 3-14 show the magnitude and phase plots of different controllers which were obtained for different α_1 , α_2 values. Figures 3-5 correspond to $\alpha_1 = 0$, $\alpha_2 = 1$, figures 6-8 correspond to $\alpha_1 = 0.1$, $\alpha_2 = 0.9$, figures 9-11 correspond to $\alpha_1 = 0.9$, $\alpha_2 = 0.1$, and figures 12-14 correspond to $\alpha_1 = 1$, $\alpha_2 = 0$. It is seen that when α_1 and α_2 are both different than zero, it is possible to achieve satisfactory tracking at steady state for both of the channels (w_1, z_1) and (w_2, z_2) . If α_1 or α_2 is zero, the corresponding channel has very poor tracking performance, due to the fact that no optimization is made with regards to that channel. It is also seen that a satisfactory tracking performance cannot be guaranteed for the frequencies other than s_0 , s_1 , and s_2 .

Since no optimization has been made concerning the transient characteristics, the transient behavior of the step responses are unsatisfactory. For example, Figure 15 shows the step responses of the transfer functions associated with the input/output pairs (w_2, z_{21}) and (w_2, z_{22}) . The overshoots are extreme for both responses and necessitate the shaping of $Q(s)$ for good transient behavior.

CONCLUDING REMARKS

We have proposed a method for the design of a controller to achieve multiple closed-loop objectives. With this method, it is possible to achieve pole placement and to minimize the norms of particularly chosen closed-loop transfer functions at steady state. This method offers an alternative to standard H_∞ and H_2 controller design methods especially for those problems where the steady-state decoupling of specific control loops is considered.

Although we have shown that the optimization problem (31) is convex, it is yet unclear how to solve that problem numerically. One possibility is to use the generic optimization programs of MATLAB. Another possibility is to develop a specific optimization algorithm tailored to this problem. The computation time may be a possible difficulty. At each iteration, the program should compute the value of the cost function. This amounts to computing the maximum values of various step responses of the closed-loop system. As the order of the $Q(s)$ increases, the number of parameters to be determined will also increase. Consequently, the computational difficulties may become more complex.

APPENDIX A-PROOFS OF TECHNICAL LEMMAS

Proof of Lemma 1. We will prove only the [If] part. The [Only If] part can be proven similarly.

Let K^* be Pareto optimal with respect to (2). If K^* is not Pareto optimal with respect to (3) then there exists K^o such that $\|f(K^o)_i\|^2 \leq \|f(K)_i\|^2$ for $i = 1, \dots, h$ where at least one of the inequalities, say $\|f(K^o)_{h_o}\|^2 \leq \|f(K)_{h_o}\|^2$, is a strict inequality. This implies that $\|f(K^o)_i\| \leq \|f(K)_i\|$ for $i = 1, \dots, h$ with $\|f(K^o)_{h_o}\| < \|f(K)_{h_o}\|$ and contradicts the fact that K^* is Pareto optimal with respect to (2). $\square$

Proof of Lemma 2. If K^* is not Pareto optimal with respect to (5) then there exists K^o such that $\phi(K^o)_i \leq \phi(K^*)_i$, $i = 1, \dots, h$ where at least one of the inequalities is a strict inequality. Since each α_i is nonnegative, this implies that $\alpha_i \phi(K^o)_i \leq \alpha_i \phi(K^*)_i$, $i = 1, \dots, h$ where at least one of the inequalities is a strict inequality. Consequently,

$$\sum_{i=1}^h \alpha_i \phi(K^o)_i < \sum_{i=1}^h \alpha_i \phi(K^*)_i,$$

which contradicts the fact that K^* is a solution of (4). $\square$

Proof of Lemma 3. We will prove only the [If] part. The [Only If] part can be proven similarly.

Let K^* be Pareto optimal with respect to (6). If $g(K^*)$ is not Pareto optimal with respect to (7) then there exists α^o satisfying $\hat{\phi}(\alpha^o)_i \leq \hat{\phi}(g(K^*))_i$, $i = 1, \dots, h$ where at least one of the inequalities is strict inequality. By the definition, this implies $\phi(g^{-1}(\alpha^o))_i \leq \phi(K^*)_i$, $i = 1, \dots, h$ where at least one of the inequalities is strict inequality. Consequently, there is a contradiction. $\square$

Proof of Lemma 4. Since $\overline{B(s)Q(s)C(s)}$ is a scalar, one can write

$$B(s)Q(s)C(s) = (C'(s) \otimes B(s)) \text{vec}(Q(s))$$

Let

$$[d_1(s) \ d_2(s) \ \dots \ d_{mn}(s)] := C'(s) \otimes B(s)$$

We will now define a collection of $mn(l+1)$ elements in $\mathcal{R}$.

For $k \in \{0, \dots, l\}$,

$$\begin{aligned} \bar{\alpha}_{1,k} &= \alpha_{1,1,k}, \dots, \bar{\alpha}_{m,k} = \alpha_{m,1,k}, \bar{\alpha}_{m+1,k} = \alpha_{1,2,k}, \bar{\alpha}_{m+2,k} = \alpha_{2,2,k}, \bar{\alpha}_{m+3,k} = \alpha_{3,2,k}, \dots, \bar{\alpha}_{2m,k} = \alpha_{m,2,k}, \\ &\dots, \bar{\alpha}_{mn-m+1,k} = \alpha_{1,n,k}, \dots, \bar{\alpha}_{mn,k} = \alpha_{m,n,k}. \end{aligned} \tag{38}$$

It holds that

$$(C'(s) \otimes B(s)) \text{vec}(Q(s)) = \sum_{i=1}^{mn} \sum_{k=0}^l \bar{\alpha}_{i,k} \frac{s^k d_i(s)}{v(s)}.$$

For $i \in \{1, \dots, mn\}$, $k \in \{0, \dots, l\}$, we let

$$f_{i,k}(t) := \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{(j\omega)^k d_i(j\omega) e^{j\omega t}}{v(j\omega) j\omega}.$$

We also define

$$a(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{a(j\omega) e^{j\omega t}}{j\omega}.$$

Then, the zero-state unit step response associated with the transfer function $A(s) + B(s)Q(s)C(s)$ is equal to

$$a(t) + \sum_{i=1}^{mn} \sum_{k=0}^l \bar{\alpha}_{i,k} f_{i,k}(t) = a(t) + \bar{\alpha} [f_{1,0}(t) \dots f_{1,l}(t) f_{2,0}(t) \dots f_{2,l}(t) \dots f_{mn,0}(t) \dots f_{mn,l}(t)]'$$

where

$$\bar{\alpha} := [\bar{\alpha}_{1,0} \dots \bar{\alpha}_{1,l} \bar{\alpha}_{2,0} \dots \bar{\alpha}_{2,l} \dots \bar{\alpha}_{mn,0} \dots \bar{\alpha}_{mn,l}]. \quad (39)$$

This completes the proof. $\square$

Proof of Lemma 6. For convenience, define $b(t) = [b_1(t) \dots b_n(t)]'$. For $\lambda \in [0, 1]$, $\alpha_1 \in \mathcal{R}^{1 \times n}$, and $\alpha_2 \in \mathcal{R}^{1 \times n}$

$$\begin{aligned} \phi(\lambda\alpha_1 + (1-\lambda)\alpha_2) &= \sup_{t \geq 0} \{a(t) + (\lambda\alpha_1 + (1-\lambda)\alpha_2)b(t)\} \\ &= \sup_{t \geq 0} \{(\lambda + (1-\lambda))a(t) + (\lambda\alpha_1 + (1-\lambda)\alpha_2)b(t)\} \\ &= \sup_{t \geq 0} \{\lambda(a(t) + \alpha_1 b(t)) + (1-\lambda)(a(t) + \alpha_2 b(t))\} \\ &\leq \sup_{t \geq 0} \{\lambda(a(t) + \alpha_1 b(t))\} + \sup_{t \geq 0} \{(1-\lambda)(a(t) + \alpha_2 b(t))\} \\ &\preceq \lambda \sup_{t \geq 0} \{a(t) + \alpha_1 b(t)\} + (1-\lambda) \sup_{t \geq 0} \{a(t) + \alpha_2 b(t)\} \\ &= \lambda\phi(\alpha_1) + (1-\lambda)\phi(\alpha_2) \end{aligned}$$

This completes the proof. $\square$

Proof of Lemma 6. For convenience, define $b(t) = [b_1(t) \dots b_n(t)]'$, $c(t) = [c_1(t) \dots c_n(t)]'$.

For any α ,

$$\phi(\alpha) = \max\{\sup_{t \geq 0} \{a_1(t) + \alpha b(t)\}, \sup_{t \geq 0} \{a_2(t) + \alpha c(t)\}, 0\} \quad (40)$$

Consider $\lambda \in [0, 1]$, $\alpha_1 \in \mathcal{R}^{1 \times n}$, and $\alpha_2 \in \mathcal{R}^{1 \times n}$.

Case 1: Assume $\phi(\lambda\alpha_1 + (1-\lambda)\alpha_2) = 0$

In this case, it is trivial to see that $\phi(\lambda\alpha_1 + (1-\lambda)\alpha_2) \leq \lambda\phi(\alpha_1) + (1-\lambda)\phi(\alpha_2)$.

Case 2: Assume

$$\phi(\lambda\alpha_1 + (1 - \lambda)\alpha_2) = \sup_{t \geq 0} \{a_1(t) + (\lambda\alpha_1 + (1 - \lambda)\alpha_2)b(t)\}.$$

In this case, Lemma 5 implies

$$\phi(\lambda\alpha_1 + (1 - \lambda)\alpha_2) \leq \lambda \sup_{t \geq 0} \{a_1(t) + \alpha_1 b(t)\} + (1 - \lambda) \sup_{t \geq 0} \{a_1(t) + \alpha_2 b(t)\} \quad (41)$$

From (40) one can write

$$\lambda \sup_{t \geq 0} \{a_1(t) + \alpha_1 b(t)\} \leq \lambda \phi(\alpha_1), \quad (1 - \lambda) \sup_{t \geq 0} \{a_1(t) + \alpha_2 b(t)\} \leq (1 - \lambda) \phi(\alpha_2). \quad (42)$$

From (41) and (42),

$$\phi(\lambda\alpha_1 + (1 - \lambda)\alpha_2) \leq \lambda \phi(\alpha_1) + (1 - \lambda) \phi(\alpha_2).$$

Case 3: Assume

$$\phi(\lambda\alpha_1 + (1 - \lambda)\alpha_2) = \sup_{t \geq 0} \{a_2(t) + (\lambda\alpha_1 + (1 - \lambda)\alpha_2)c(t)\}$$

In a similar way to Case 2, it can be shown that

$$\phi(\lambda\alpha_1 + (1 - \lambda)\alpha_2) \leq \lambda \phi(\alpha_1) + (1 - \lambda) \phi(\alpha_2).$$

This completes the proof. $\square$

APPENDIX B-SOFTWARE IMPLEMENTATION

We describe the software implementation of the steps 1 and 2 of the procedure.

The system is created in XMATH using `SystemBuild`. The mathematical model of the state-space system is generated in XMATH using the script function `parti7.ms`. The data needed for the optimization problems (35), (36), and (37) is transferred to MATLAB via an executable program called `conv`. The source code of `conv` is a C program called `col.c`. The MATLAB program `cono3.m` defines various variables to be used with the MATLAB programs `q1gen.m`, `q2gen.m`, and `q3gen.m` which solve the convex optimization problems (35), (36), and (37), respectively. The MATLAB program `interpol3.m` determines a stable rational matrix $Q(s)$ satisfying $Q(s_i) = Q^*(s_i)$, $i = 0, 1, 2$, where $Q^*(s_0)$, $Q^*(s_1)$, and $Q^*(s_2)$ are the solutions of (35), (36), (37), respectively. It also converts the $Q(s)$ data to a XMATH readable format. The XMATH file `oku.ms` generates the controller $K(s) = (T_X(s) - Q(s)T_N(s))^{-1}(T_Y(s) - Q(s)T_M(s))$ based on $Q(s)$. Finally, this controller is substituted in a simulation block in `SystemBuild` and the simulations are performed. The following pages include the printouts of programs `parti7.ms`, `col.c`, `cono3.m`, `q1gen.m`, `q2gen.m`, `q3gen.m`, `interpol3.m`, `oku.ms` and the MATLAB subroutines `stac.m` and `ogmen.m`.

parti7.ms

col.c

cono3.m

q1gen.m

q2gen.m

q3gen.m

interpol3.m

oku.ms

stac.m

ogmen.m

APPENDIX C-FIGURES

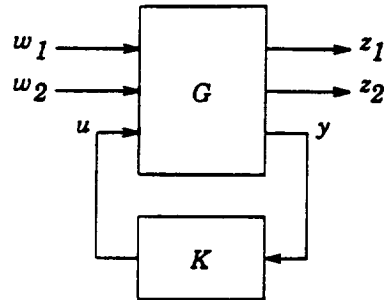


Figure 1: Multiobjective control feedback structure.

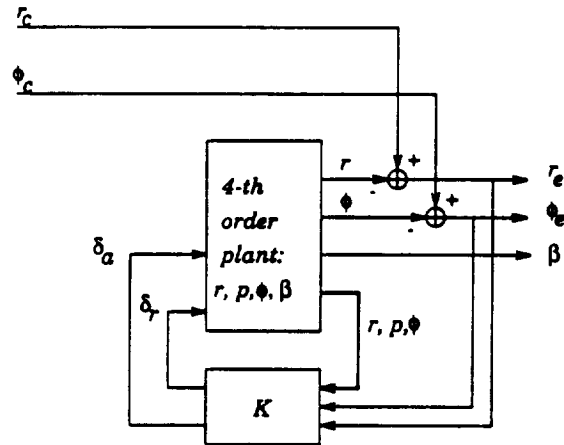


Figure 2: Closed-loop system of the example.

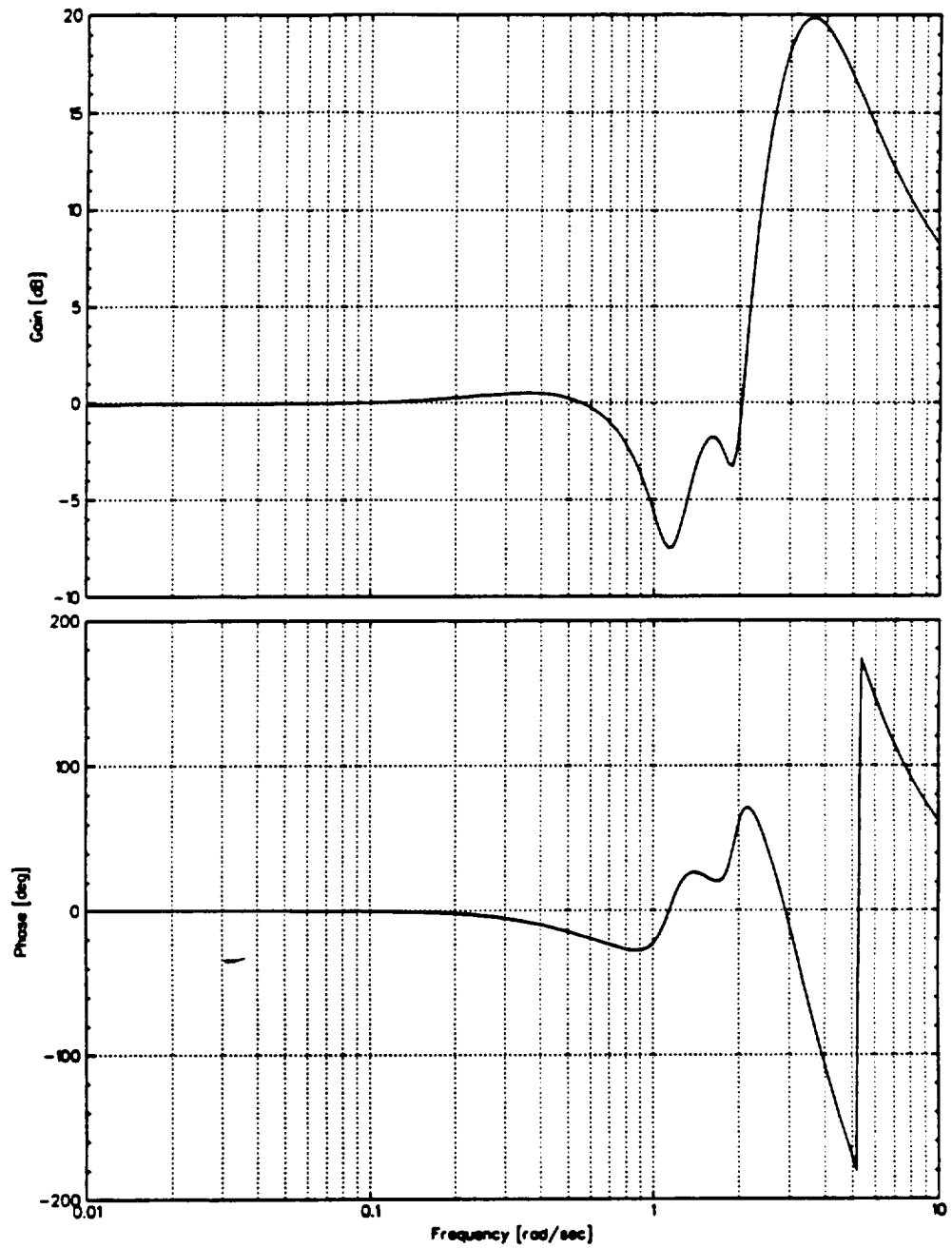


Figure 3: Magnitude and phase plots of the transfer function from w_1 to z_1 . ($\alpha_1 = 0$, $\alpha_2 = 1$.)

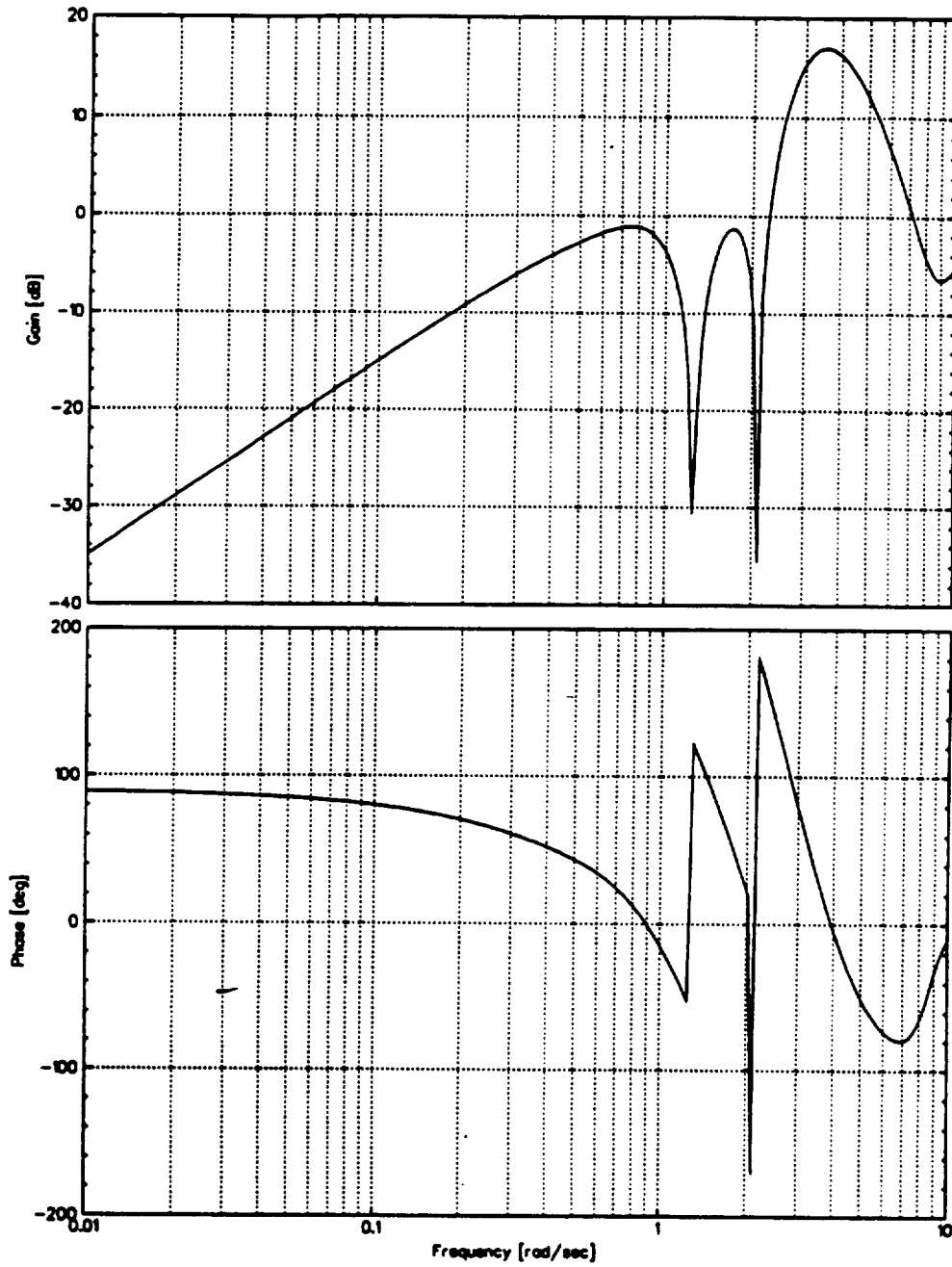


Figure 4: Magnitude and phase plots of the transfer function from w_2 to z_{21} . ($\alpha_1 = 0$, $\alpha_2 = 1$.)

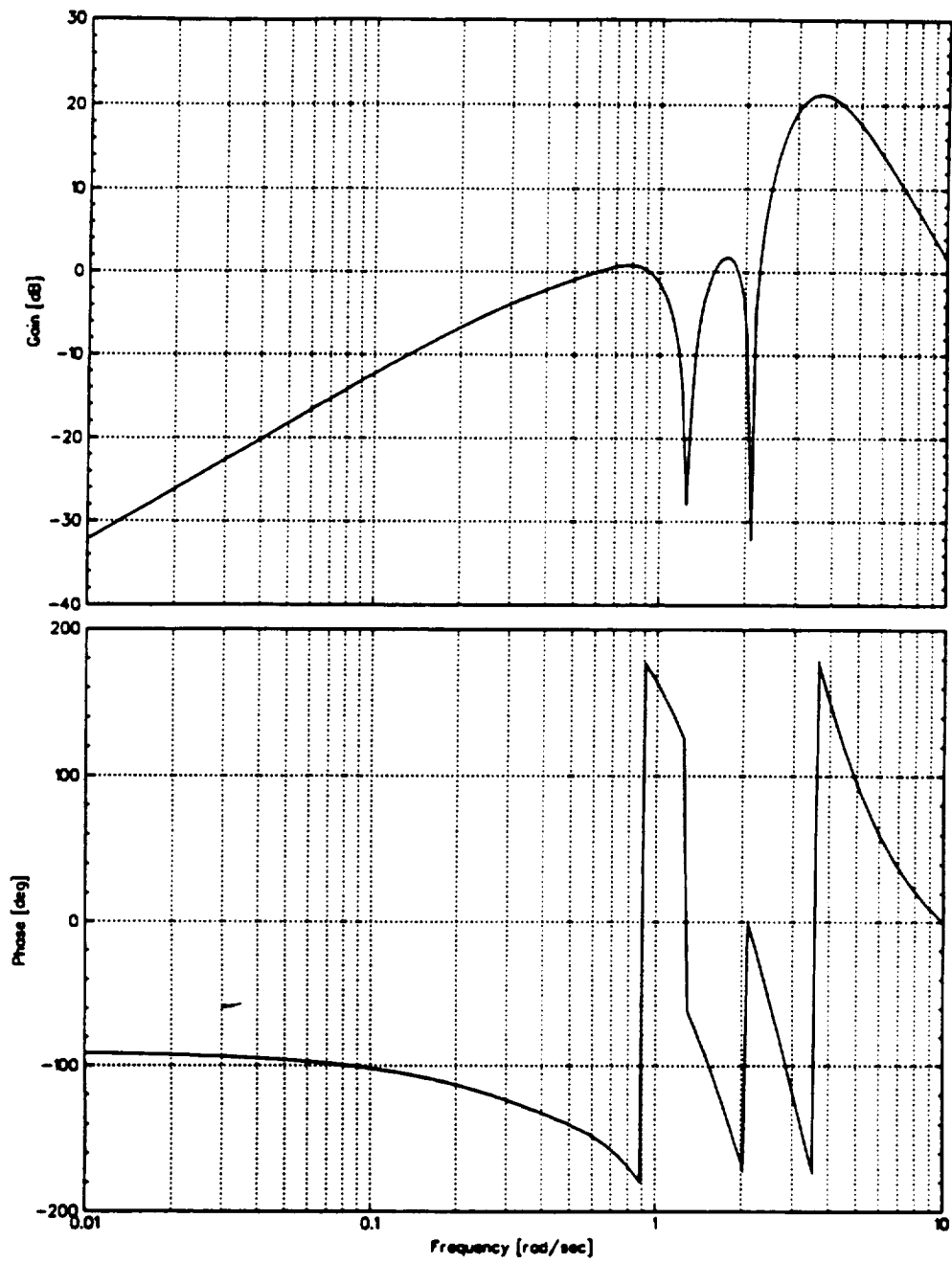


Figure 5: Magnitude and phase plots of the transfer function from w_2 to z_{22} . ($\alpha_1 = 0$, $\alpha_2 = 1$.)

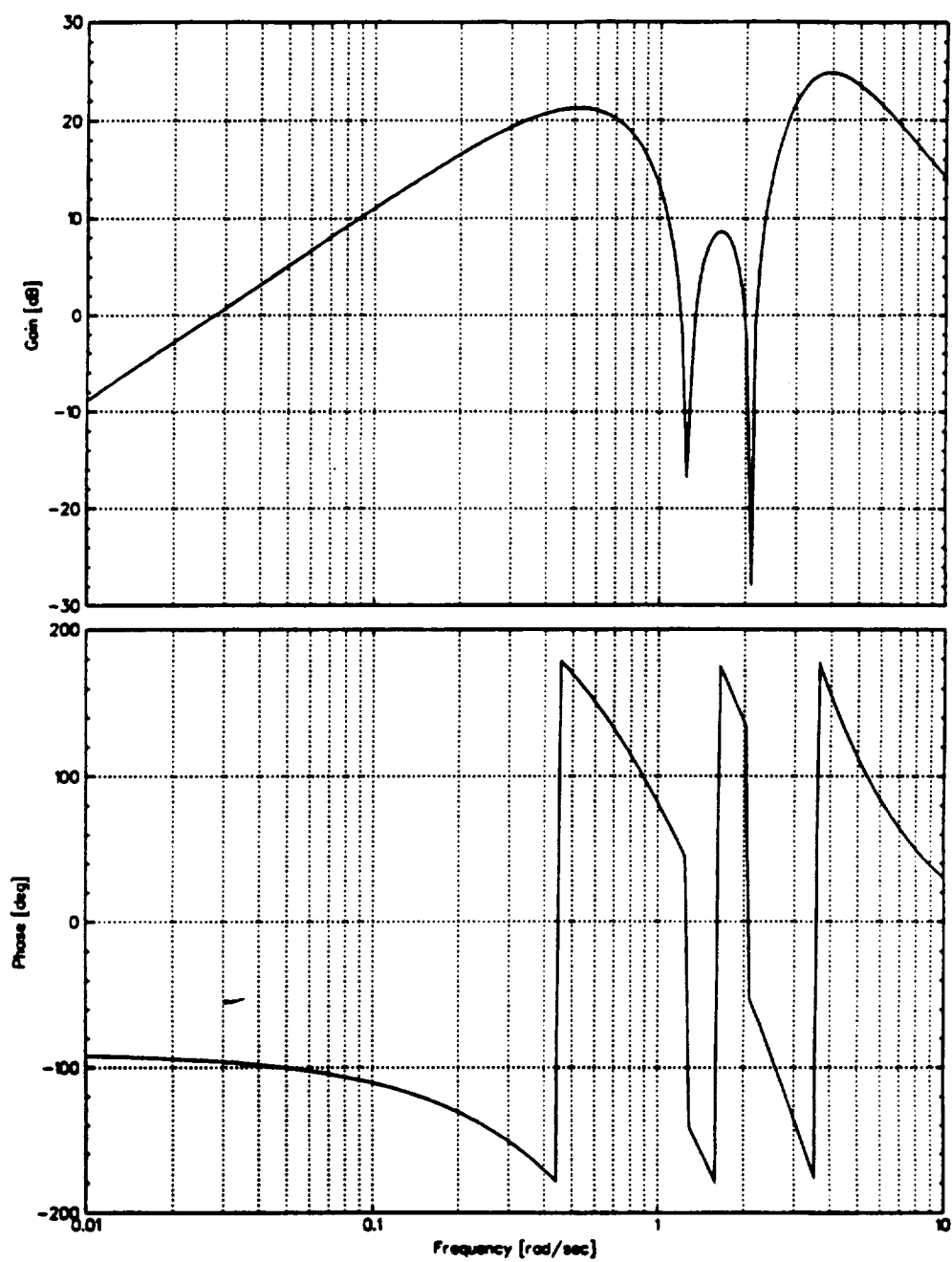


Figure 6: Magnitude and phase plots of the transfer function from w_1 to z_1 . ($\alpha_1 = 0.1$, $\alpha_2 = 0.9$.)

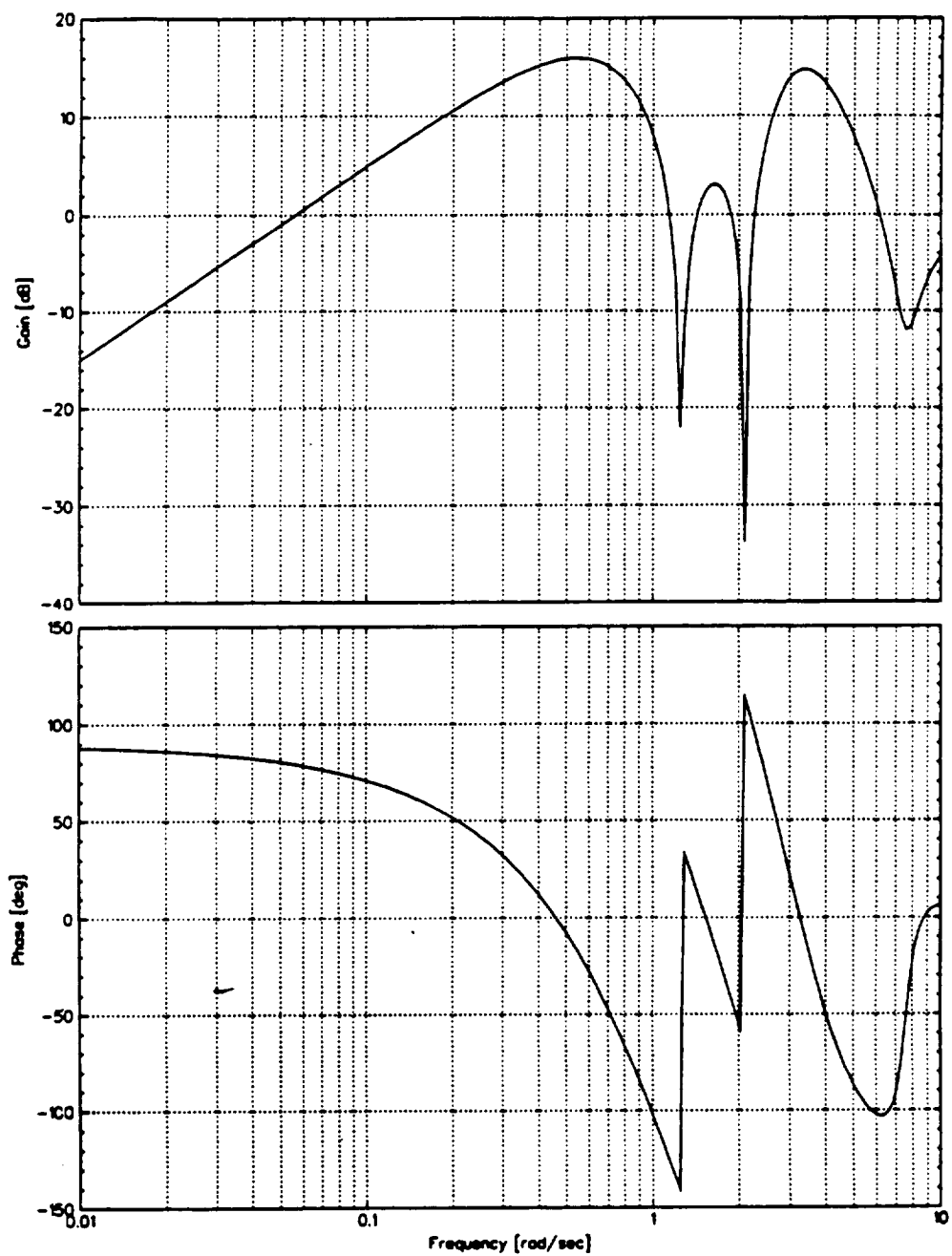


Figure 7: Magnitude and phase plots of the transfer function from w_2 to z_{21} . ($\alpha_1 = 0.1$, $\alpha_2 = 0.9$.)

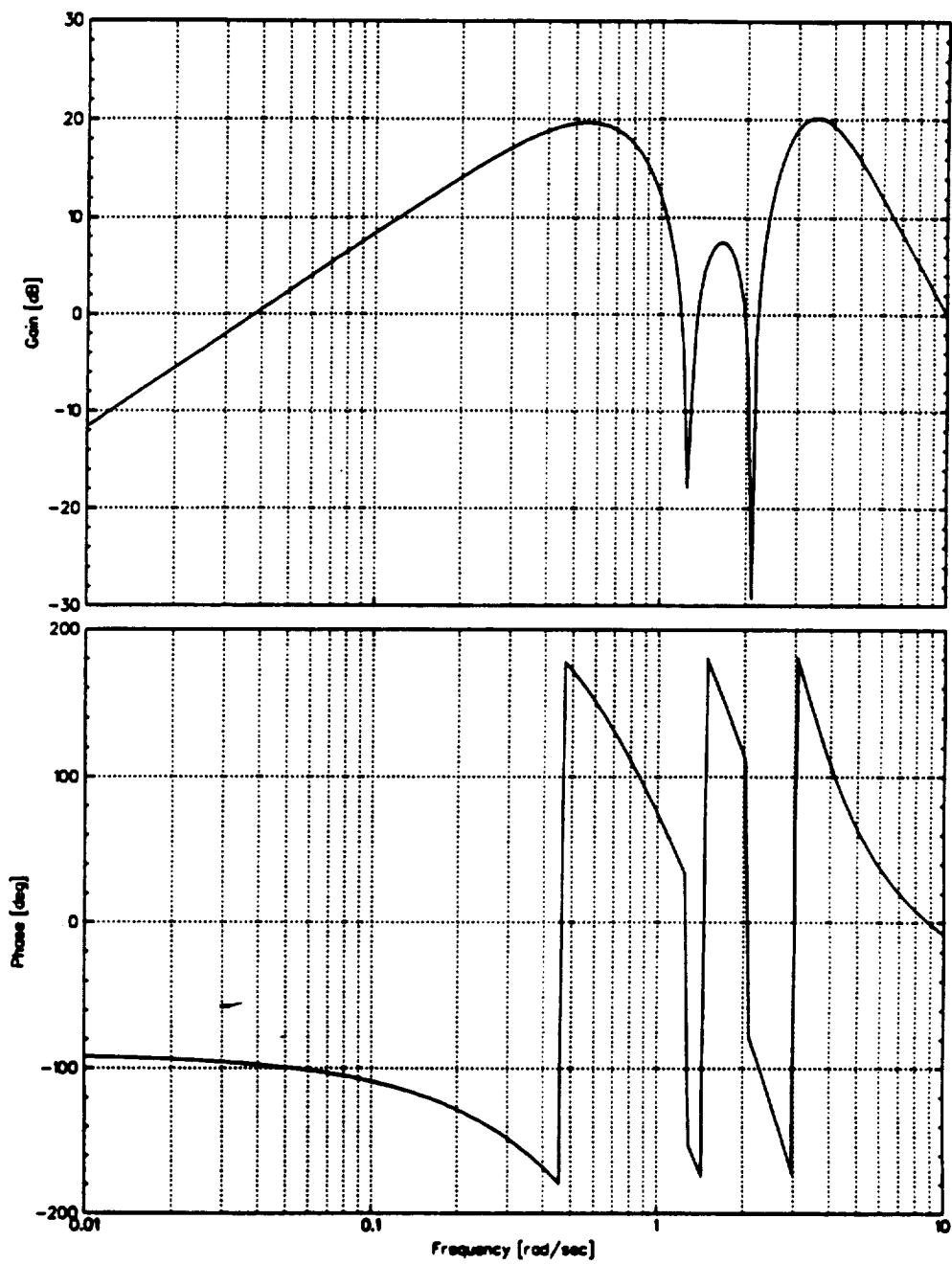


Figure 8: Magnitude and phase plots of the transfer function from w_2 to z_{22} . ($\alpha_1 = 0.1$, $\alpha_2 = 0.9$.)

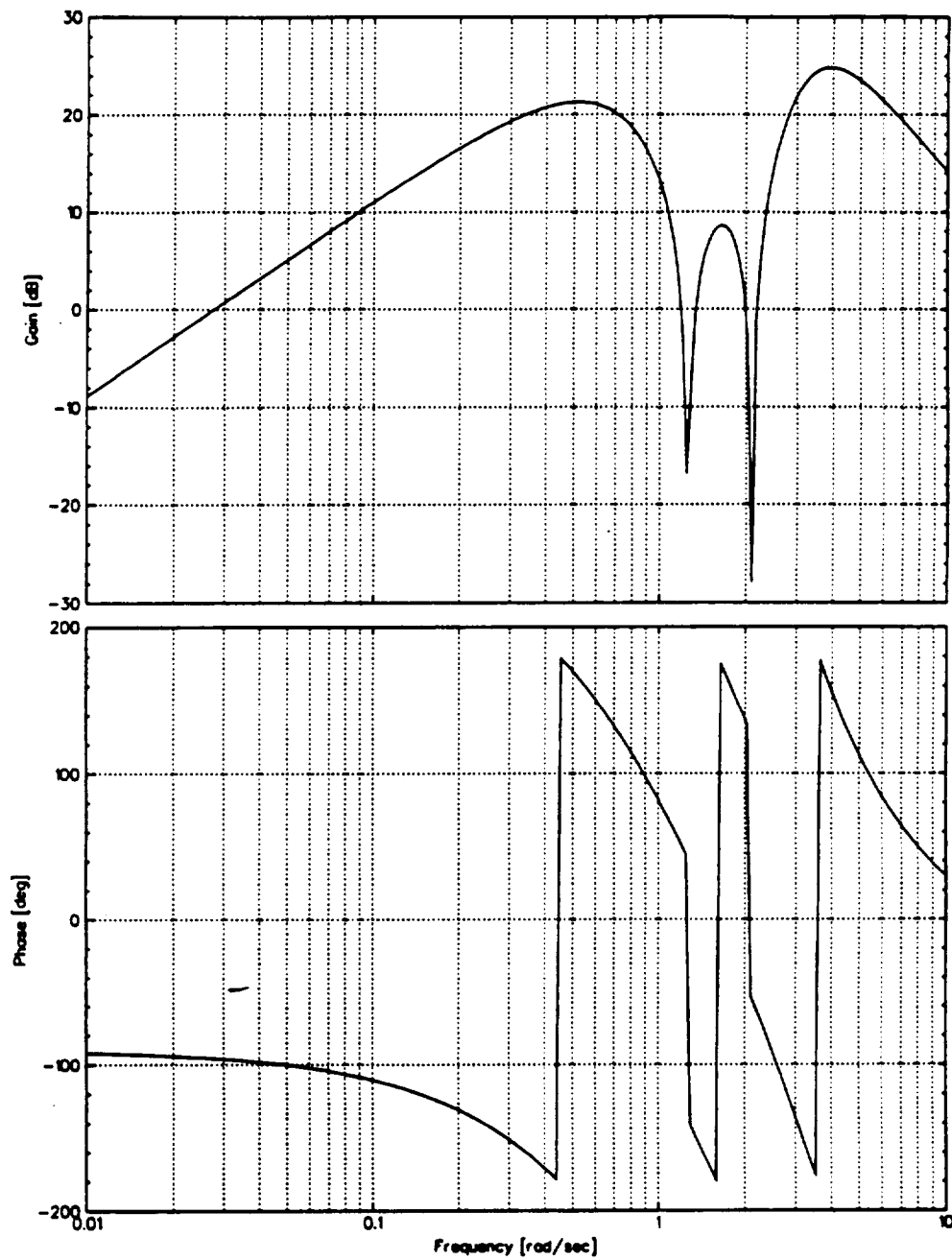


Figure 9: Magnitude and phase plots of the transfer function from w_1 to z_1 . ($\alpha_1 = 0.9$, $\alpha_2 = 0.1$.)

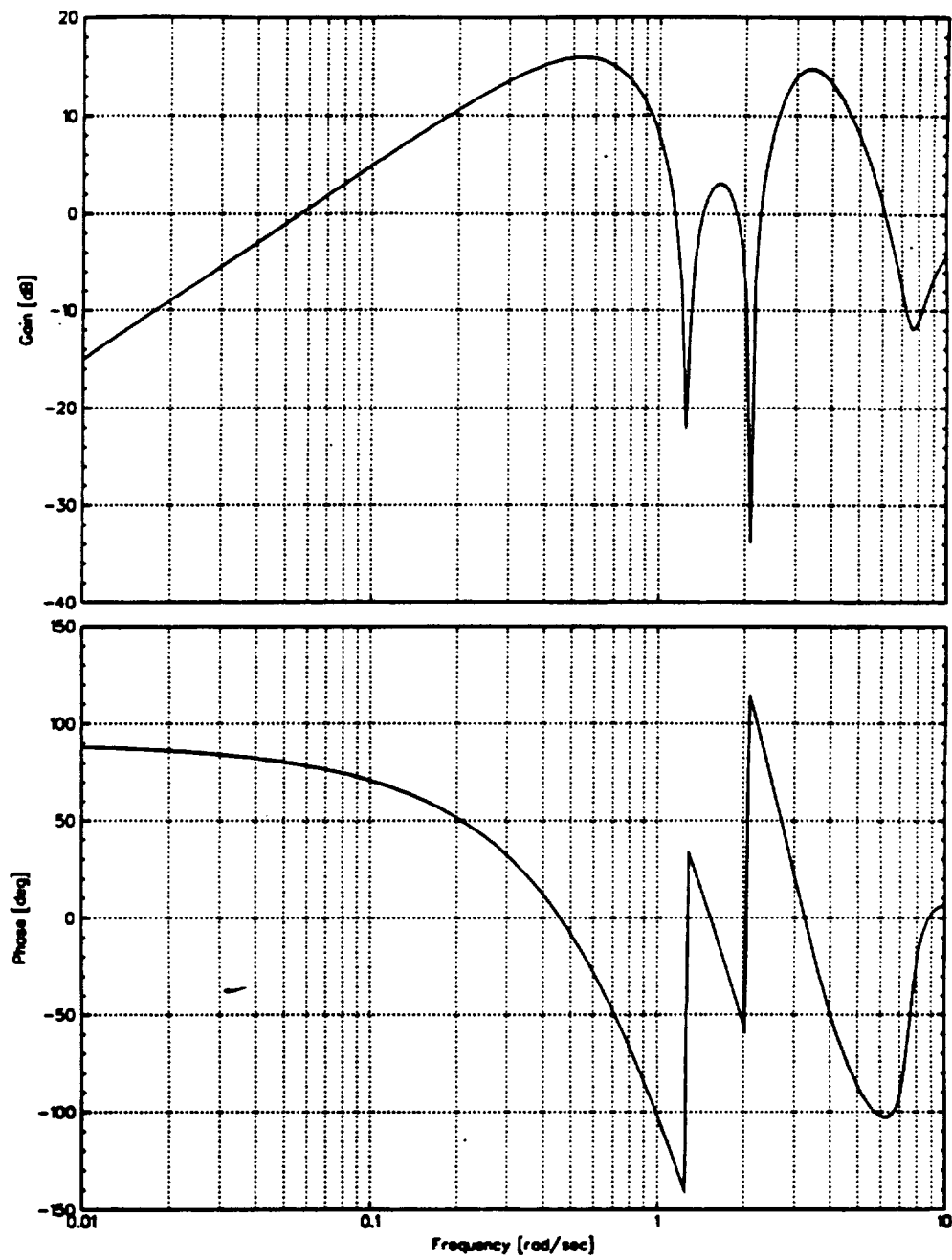


Figure 10: Magnitude and phase plots of the transfer function from w_2 to z_{21} . ($\alpha_1 = 0.9$, $\alpha_2 = 0.1$.)

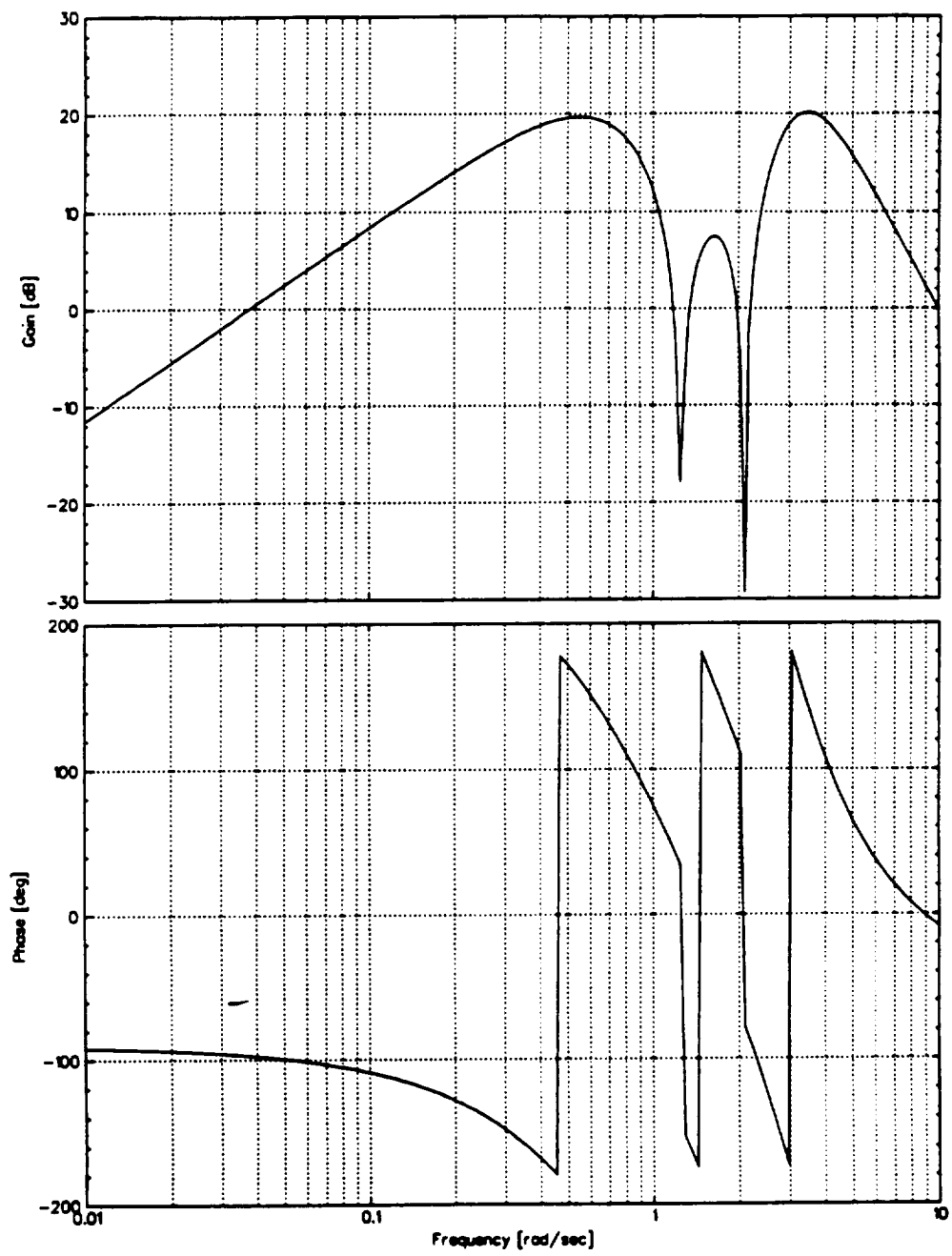


Figure 11: Magnitude and phase plots of the transfer function from w_2 to z_{22} . ($\alpha_1 = 0.9$, $\alpha_2 = 0.1$.)

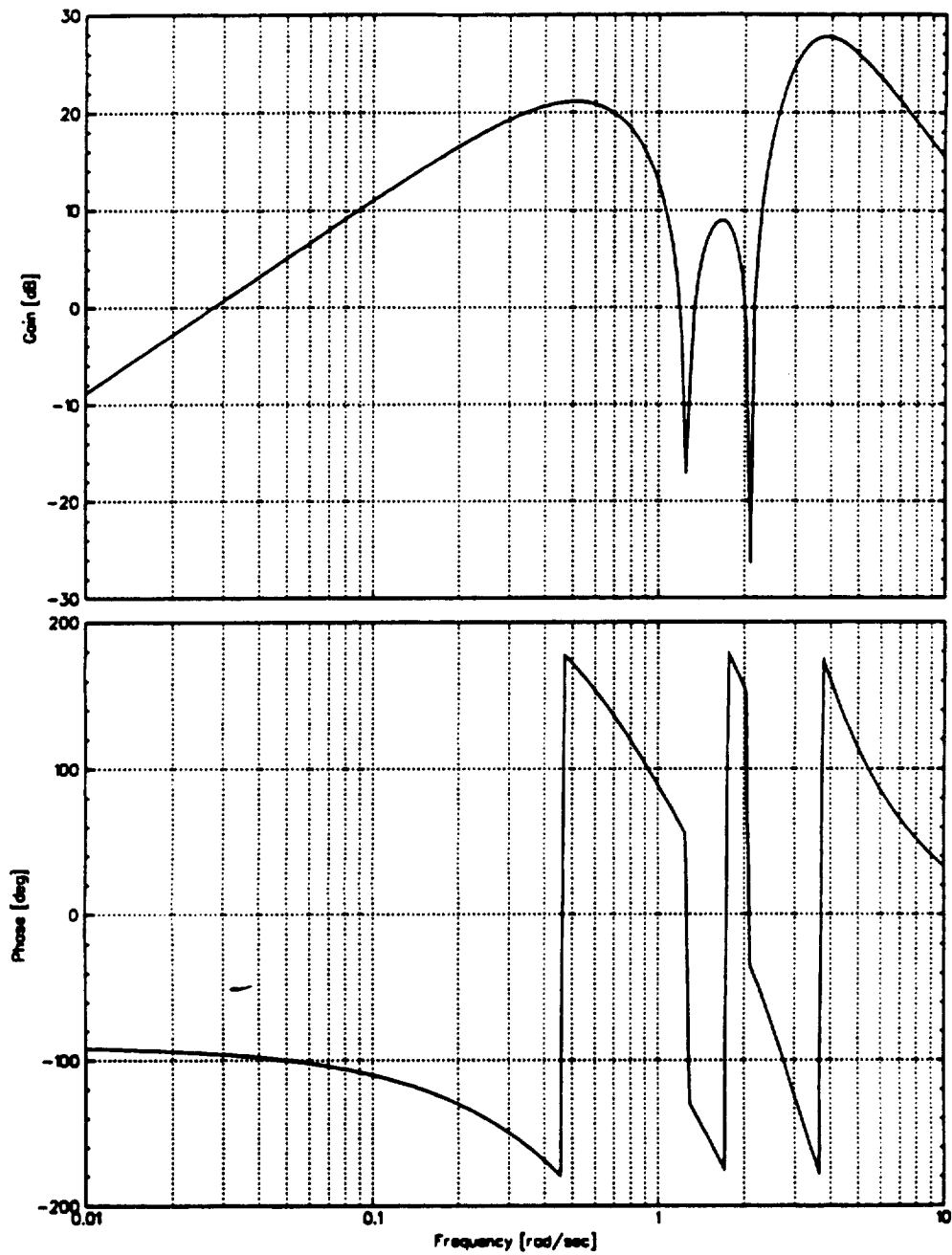


Figure 12: Magnitude and phase plots of the transfer function from w_1 to z_1 . ($\alpha_1 = 1$, $\alpha_2 = 0$.)

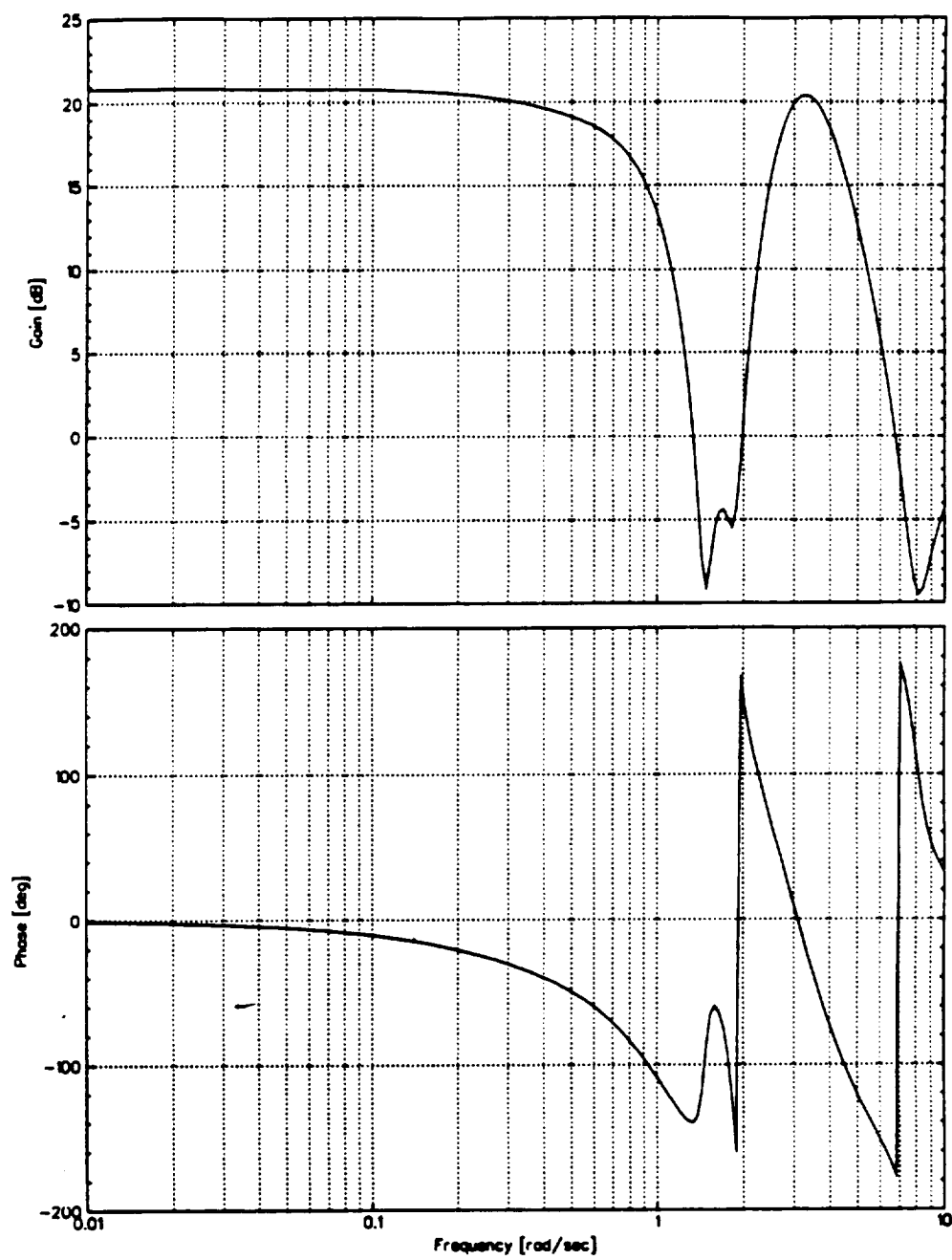


Figure 13: Magnitude and phase plots of the transfer function from w_2 to z_{21} . ($\alpha_1 = 1$, $\alpha_2 = 0$.)

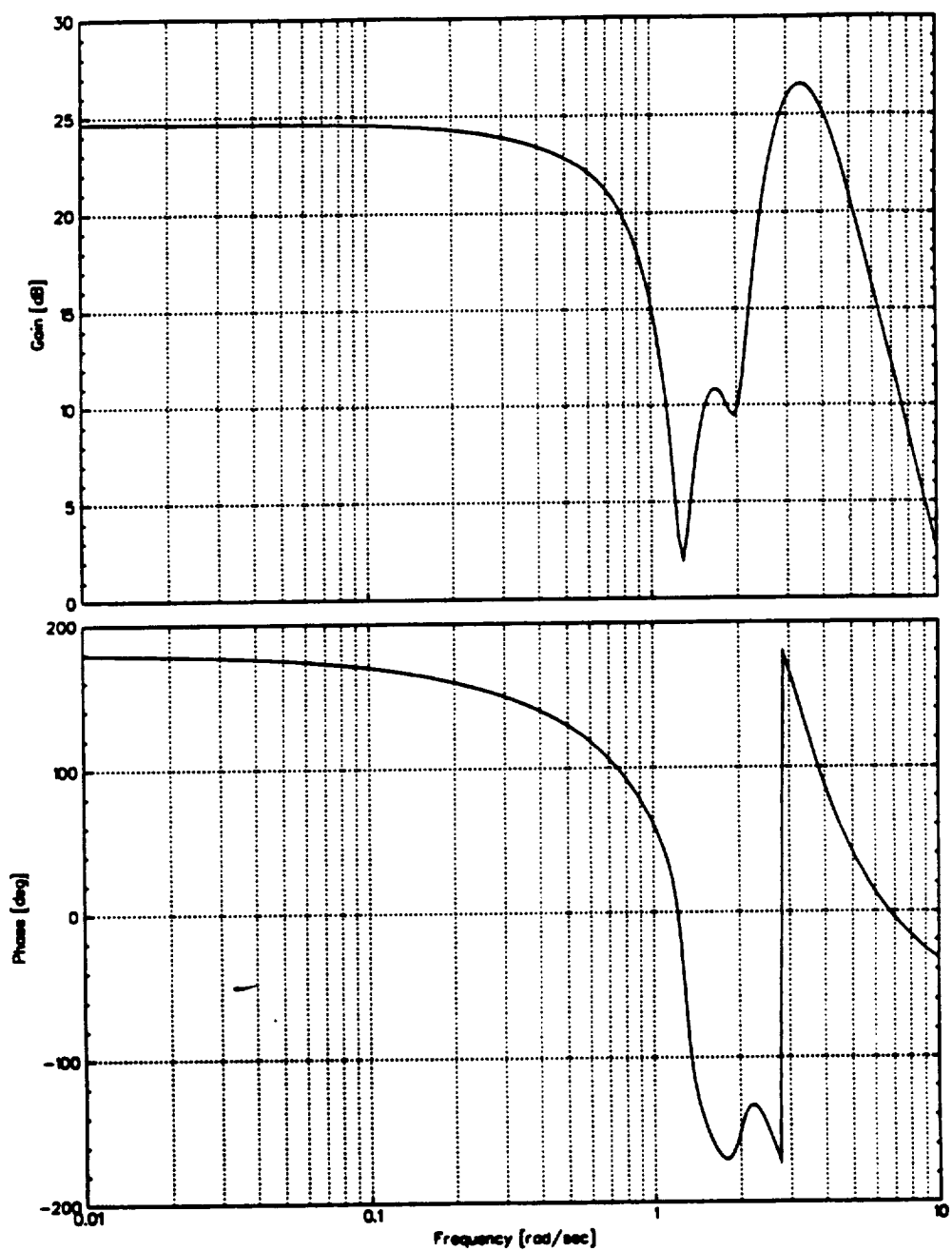


Figure 14: Magnitude and phase plots of the transfer function from w_2 to z_{22} . ($\alpha_1 = 1$, $\alpha_2 = 0$.)

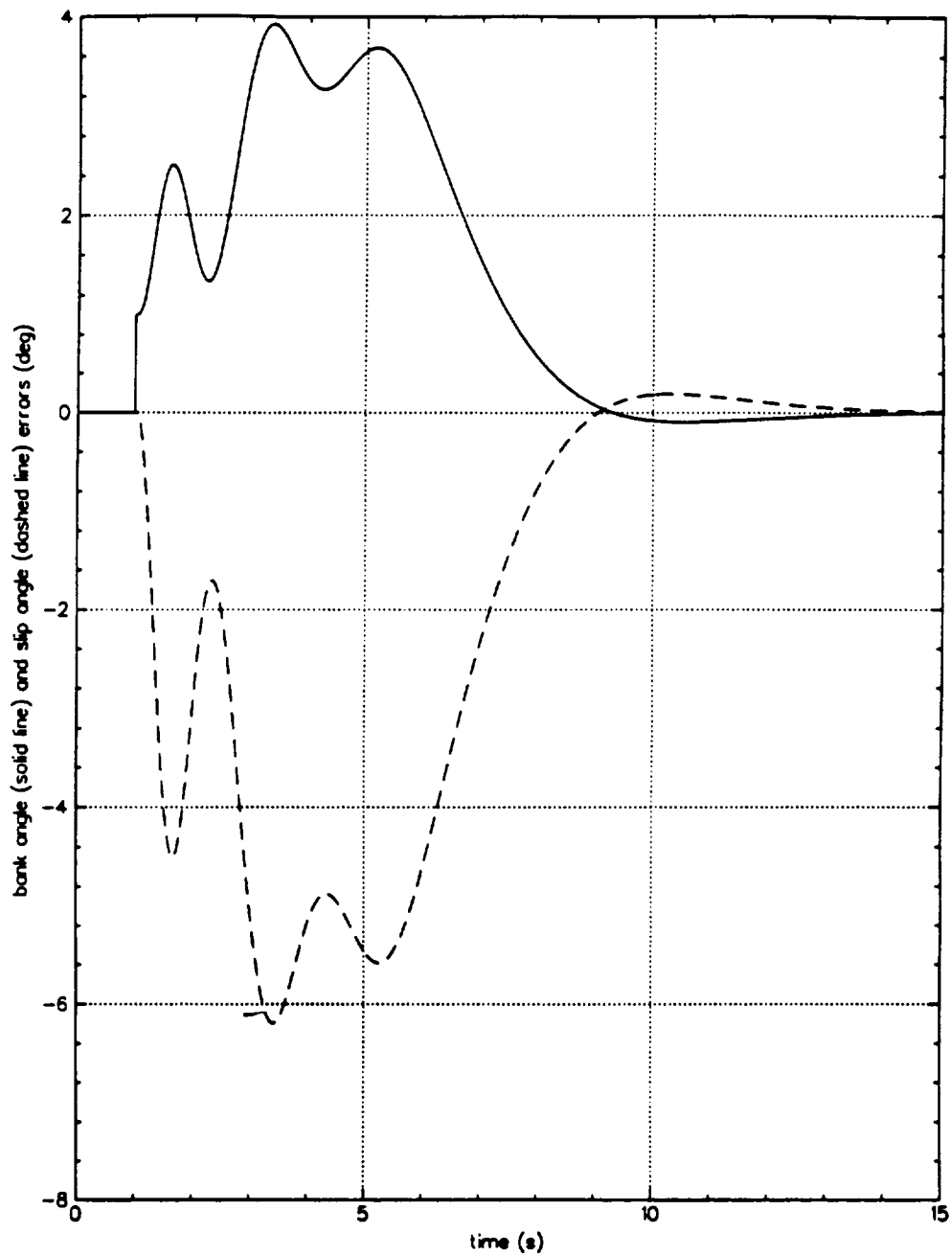


Figure 15: Step responses of the transfer functions from w_2 to z_{21} (solid line) and z_{22} (dashed line). In this particular example, the solid line represents the bank angle error and the dashed line represents the slip angle error in degrees. ($\alpha_1 = 0$, $\alpha_2 = 1$.)

Computational Methods for Strongly Stabilizing $\mathcal{H}^\infty$ Controllers

1 Introduction

In this report several research directions are described for developing computational methods to obtain stable $\mathcal{H}^\infty$ controllers for aircraft dynamics. The proposed techniques are based on the results of the PI's research performed for the NASA grant no. NAG3-1723.

Recall that an important motivation behind stable (i.e. strongly stabilizing) controller design is reliability against faults in the measurements. Also, from the implementation point of view, it is practically impossible to *test* an unstable controller in open loop. A necessary and sufficient condition for the existence of a strongly stabilizing controller is the parity interlacing property (p.i.p.) [1, 2]. There are procedures for constructing stable controllers which stabilize a given plant, [3, 1, 2]. But the problem of finding a (sub)optimal one, in the sense of $\mathcal{H}^\infty$, is currently open. Some promising results appear in [4] (see also references therein) on the $\mathcal{H}^2$ version of this problem. The effects of weight selection on the stability of the optimal $\mathcal{H}^\infty$ controller for SISO plants have been studied in [5].

The results of [6], [7] and [8] can be used in order to obtain a parametrization of all suboptimal $\mathcal{H}^\infty$ controllers. Most commercially available softwares (e.g. robust control modules of *MATLAB* and *MATRIX*) generate the so-called "central controller" of [7]. This research is about finding a stable controller in the parametrization of all suboptimal $\mathcal{H}^\infty$ controllers. For a given admissible $\mathcal{H}^\infty$ suboptimal performance level the central controller may be unstable, but there may be stable controllers in the set of all suboptimal controllers.

The rest of this report is organized as follows. In the next section stable $\mathcal{H}^\infty$ controller design problem is defined and current research findings are summarized. Then, in Section 3 an algorithm for interpolation with outer functions is described in connection with stable controller design. Another approach, which uses genetic algorithms (GAs), is described in Section 4. The search algorithms reported in Sections 3 and 4 are promising research directions determined in the project NAG3-1723. This is an ongoing research with a continuing support from NASA Lewis Research Center (under a new grant no. NAG3-1826).

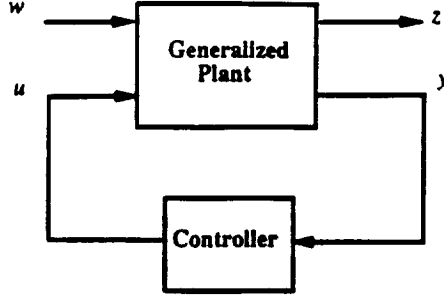


Figure 1: Standard Feedback System

2 Stable $\mathcal{H}^\infty$ controller design

2.1 Standard problem set-up

The so-called “standard $\mathcal{H}^\infty$ control problem” deals with the system shown in Figure 1. The system equations are assumed to be given by the following

$$\dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t) \quad (1)$$

$$z(t) = C_1x(t) + D_{11}w(t) + D_{12}u(t) \quad (2)$$

$$y(t) = C_2x(t) + D_{21}w(t) + D_{22}u(t) \quad (3)$$

where x represents combined states (states of the plant and the weights) in the system, and components of w are the exogenous signals (reference inputs, disturbances, measurement noises). The optimal $\mathcal{H}^\infty$ problem is to find a feedback controller K (whose input is y and output u) so that closed loop system is stable, and the worst energy amplification from w to z is minimized. This problem is equivalent to finding a stabilizing controller which minimizes $\|T_{zw}\|_\infty$, where $T_{zw}(s)$ is the closed loop transfer function from w to z . The suboptimal $\mathcal{H}^\infty$ control problem is to find a stabilizing controller so that $\|T_{zw}\|_\infty < \gamma$, for a specified performance level γ .

2.2 Controller formulae

The formulae of [6, 7, 8] for $\mathcal{H}^\infty$ controllers, which satisfy a certain specified performance level γ , is given by

$$\dot{x}_c = A_c x_c + B_{1c} y + B_{2c} q$$

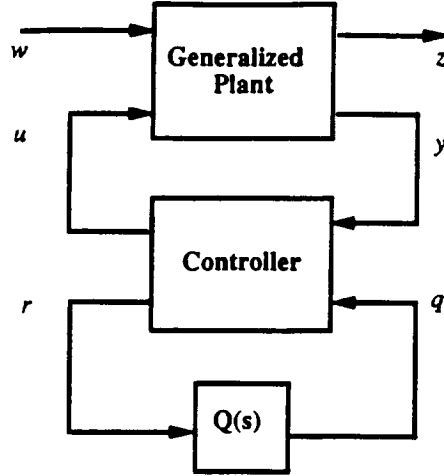


Figure 2: Suboptimal $\mathcal{H}^\infty$ controllers

$$\begin{aligned} u &= C_{1c}x_c + \hat{D}_{11}y + \hat{D}_{12}q \\ r &= C_{2c}x_c + \hat{D}_{21}y \end{aligned}$$

where $A_c, B_{1c}, B_{2c}, C_{1c}, C_{2c}, \hat{D}_{11}, \hat{D}_{12}, \hat{D}_{21}$ are computed from two algebraic Riccati equations, and $q(t)$ is generated by a transfer matrix $Q(s)$, whose input is $r(t)$, with $Q \in \mathcal{H}^\infty$ (that is Q must be stable) and $\|Q\|_\infty < \gamma$. Obviously, there are infinitely many choices for $Q(s)$ and hence there are infinitely many suboptimal controllers. Implementation of this controller is shown in Figure 2.

In particular, one can choose $Q(s) = 0$, this gives the “central” controller. Note that the central controller is stable if and only if A_c is a stable matrix (i.e. all the eigenvalues of A_c have strictly negative real parts). Whereas, stability of a controller which is obtained from a non-zero $Q(s)$ depends on $\bar{A}_c, B_{2c}, C_{2c}$ and $Q(s)$.

2.3 A BMI approach

Note that if a state space realization of Q is taken as

$$\begin{aligned} \dot{x}_q &= A_q x_q + B_q r \\ q &= C_q x_q + D_q r \end{aligned}$$

then the overall $\mathcal{H}^\infty$ controller is

$$\begin{aligned} \begin{bmatrix} \dot{x}_c \\ \dot{x}_q \end{bmatrix} &= \underbrace{\begin{bmatrix} A_c + B_{2c}D_qC_{2c} & B_{2c}C_q \\ B_qC_{2c} & A_q \end{bmatrix}}_{A_K} \begin{bmatrix} x_c \\ x_q \end{bmatrix} + \underbrace{\begin{bmatrix} B_{1c} + B_{2c}D_q\hat{D}_{21} \\ B_q\hat{D}_{21} \end{bmatrix}}_{B_K} y \\ u &= \underbrace{\begin{bmatrix} C_{1c} + \hat{D}_{12}D_qC_{2c} & \hat{D}_{12}C_q \end{bmatrix}}_{C_K} \begin{bmatrix} x_c \\ x_q \end{bmatrix} + \underbrace{(\hat{D}_{11} + \hat{D}_{12}D_q\hat{D}_{21})}_{D_K} y \end{aligned}$$

The $\mathcal{H}^\infty$ controller is stable if and only if

- (i) A_K is stable, i.e. all eigenvalues of A_K are in the open left half plane, and
- (ii) $Q(s)$ parametrizing the controller is stable (i.e. A_q is stable), and $\|Q\|_\infty < \gamma$.

These conditions are equivalent to the following.

- (i) A_K is stable if and only if there exists P such that $P = P^T > 0$, and

$$A_K^T P + P A_K < 0$$

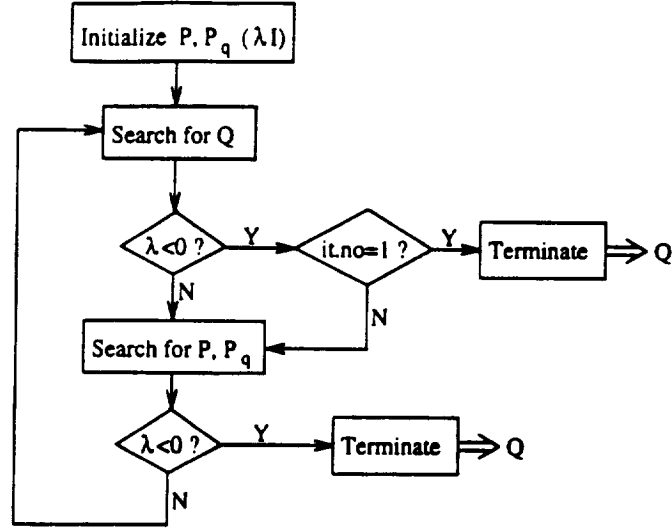
- (ii) Q is stable with $\|Q\|_\infty < \gamma$ if and only if there exists P_q such that $P_q = P_q^T > 0$, and

$$A_q^T P_q + P_q A_q - (P_q B_q + C_q^T D_q) R^{-1} (P_q B_q + C_q^T D_q)^T < 0, \text{ where } R = D_q^T D_q - \gamma^2 I < 0$$

or equivalently

$$\begin{bmatrix} A_q^T P_q + P_q A_q & P_q B_q & C_q^T \\ B_q^T P_q & -I & D_q^T \\ C_q & D_q & -\gamma^2 I \end{bmatrix} < 0.$$

Using these equivalent conditions the following bilinear matrix inequality (BMI) optimization problem can be posed for finding stable $\mathcal{H}^\infty$ controllers:



$$\begin{aligned}
 \min \quad & \lambda \\
 \text{subject to} \quad & A_K^T P + P A_K - \lambda I \leq 0 \\
 & 0 \leq P + \lambda I \\
 & \begin{bmatrix} A_q^T P_q + P_q A_q & P_q B_q & C_q^T \\ B_q^T P_q & -I & D_q^T \\ C_q & D_q & -\gamma^2 I \end{bmatrix} - \lambda I \leq 0 \\
 & 0 \leq P_q + \lambda I \\
 & 0 \leq \lambda + 1
 \end{aligned}$$

If there exists $\lambda < 0$ the problem is solved. An alternating optimization method is used with the LMI (linear matrix inequality) toolbox of MATLAB to search for feasible solutions to this optimization problem. Note that for P and P_q fixed we have LMIs in the variable Q ; similarly when Q is fixed we have LMIs in the variables P and P_q . A flowchart of the proposed search is given above. This algorithm has been tested on numerical examples for aircraft control problems (tracking and gust alleviation problems defined in [9]). Depending on the realization of the central controller, the algorithm may or may not find a feasible stable $\mathcal{H}^\infty$ controller. Optimal P and P_q found from this approach tend to be ill conditioned for the numerical examples considered in this study.

The BMI optimization method described above tries to find a stable $\mathcal{H}^\infty$ controller as follows: first some arbitrary positive definite P and P_q are chosen, then λ is minimized, and the corresponding Q is determined, if $\lambda < 0$ we have a solution. In the next step, for the Q found before, we search for P and P_q , and check $\lambda < 0$, if not, for this P and P_q find a new Q , and iterate until a feasible solution is found. Obviously, there is no guarantee that this program will terminate with a feasible solution. A similar method which uses coupled LMIs is also proposed in [12]. But their method is also conservative and fails to find a solution for the aircraft control problems studied here.

Solutions to BMI optimization problems are currently investigated by several researchers see e.g. [13] [14] and references therein. However, an efficient solution procedure has not been found yet. In fact it has been shown that such problems are NP-hard [15], meaning that it is rather unlikely to find a polynomial time solution. Hence one is restricted to conservative search techniques like the one proposed above.

3 Interpolation with outer functions

The structure of A_K given above shows that the $\mathcal{H}^\infty$ controller is stable if and only if Q is a stable “controller,” with $\|Q\|_\infty < \gamma$, stabilizing the “plant” $G_C := (A_c, B_{2c}, C_{2c})$. Given G_C , all stable controllers Q stabilizing G_C can be parameterized by finding all outer (minimum phase) functions satisfying certain interpolation conditions. See [10] and [11] for the details. In this research a Matlab based program will be coded to generate this parameterization. Then, an optimization will be performed on the free parameter to find a feasible Q , which satisfies $\|Q\|_\infty < \gamma$.

Key steps to be followed are as below. For simplicity the SISO case is described here, the MIMO case will be considered in the actual research. Let $p_1, \dots, p_\ell$ be the right half plane poles of G_C and $z_1, \dots, z_k$ be the right half plane zeros of G_C . Then, $G_C = m_n C_o / m_d$ where m_n is inner with zeros z_i 's, m_d is inner with zeros p_j 's and C_o is outer. When Q is stable and Q stabilizes G_C we have $S_C := (1 + G_C Q)^{-1} = m_d S_o$ for some outer S_o such that

$$S_o(z_i) = 1/m_d(z_i) \quad i = 1, \dots, k.$$

Once such S_o is found, we can compute Q as

$$Q = \frac{S_o^{-1} - m_d}{m_n C_o}.$$

But for the solution of stable $\mathcal{H}^\infty$ control problem one also needs $\|Q\|_\infty < \gamma$. Therefore, the problem is reduced to finding an outer function S_o such that

- (i) $S_o(z_i) = 1/m_d(z_i)$ for $i = 1, \dots, k$ and
- (ii) $\|C_o^{-1}(S_o^{-1} - m_d)\|_\infty < \gamma$.

In [10] and [11] the problem of finding S_o satisfying (i) and having $\|S_o\|_\infty < \rho$ has been studied and all solutions are parameterized by appropriately modifying the usual Nevanlinna-Pick algorithm. Now, a solution to stable $\mathcal{H}^\infty$ control problem can be investigated by:

Task 1. implementing the above mentioned parameterization for a large ρ , and

Task 2. searching for an element which satisfies the condition (ii), from this parameterization.

4 Genetic algorithms for stable $\mathcal{H}^\infty$ controller search

4.1 General description

Genetic algorithms (GAs) are a class of heuristic search methods, just like simulated annealing. GAs borrow ideas from the mechanisms of evolution and natural genetics.

Genetic Algorithms are inspired by the natural search and selection processes leading to the survival of the fittest individual. They are stochastic search processes directed toward increasing the fitness of an individual, and unlike some gradient search techniques which may get stuck in local solutions, due to their stochastic nature GAs can locate the globally optimal solutions. GAs borrow some terminology from biology to describe the elements of a GA. Beginning from the most basic, the elements that make up the setting for GA are:

- GENE : A gene is a single digit number. Depending on the base in which the numbers are described, it is either an element of the set $\{0, 1, \dots, 9\}$ (Decimal representation), or the set $\{0, 1\}$ (Binary representation).

- **CHROMOSOME** : A chromosome is a string of genes;

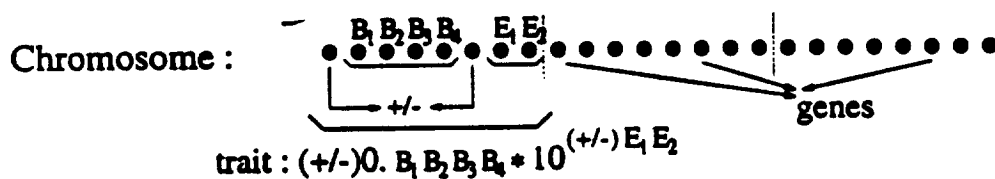
Ex : 1232378682, 3247866820012, etc. (Decimal rep.)

1001011010, 101101011011, etc. (Binary rep.)

Chromosomes are encoded forms of the parameters (matrices, vectors) that the GA is searching for. Each single chromosome is a candidate parameter collection, which can solve the optimal search problem.

- **TRAIT** : A trait is a decimal number. When chromosomes are decoded, each entry of the parameter matrices is called a trait.
- **MEMBER** : A member(individual) is the object that GA is trying to solve for. It is represented by its chromosome. Chromosome determines a member's traits, which in turn determine the member.
- **PARENT** : A parent is a member which participates in the creation of a new member. (The operations performed to get the new member will be discussed later.)
- **POPULATION** : A population is a set of members. During its operation GA works on a set of candidates, which are the members of the population.
- **FITNESS** : Fitness is the measure of suitability of the members. To find the optimal solution to the search, the objective of GA is to maximize the fitness, i.e. optimal solution of the search is the element which is found to have the largest fitness value.

Coding Scheme :(for decimal representation)



The fundamental mechanisms of GA are inspired from the theory of evolution. The three fundamental mechanisms are:

- **CROSSOVER** : With a predetermined probability p_c two members interchange digits from their chromosomes, creating two new members. The idea is to carry the good features of some members to the others, and create “fitter” members. Typically p_c is a large number like 0.85 or 0.9.
- **MUTATION** : With a predetermined probability p_m the digits of resulting chromosomes after crossover are arbitrarily altered. This operation allows creating members with totally different characteristics, and is the main mechanism which prevents getting stuck at local optimums. Typically p_m is chosen to be a small number like 0.2 or 0.25, as we don’t want to have a totally random search.
- **ELITISM** : This operation carries the fittest member from the previous generation to the next generation. It prevents the fittest member from getting lost due to crossover and mutation operations.

Genetic Algorithm Operation :

The algorithm starts with a population of candidate parameters, which can be totally random, or some of which, if available, can be suboptimal previously found solutions to the problem. The method requires a fitness function to be chosen, which is to be maximized by the desired parameter vector. The members having higher fitnesses get a higher chance in participating in the creation of the new generation. While choosing the parents the so called “roulette wheel” is used, in which the probability of each member being a parent is directly proportional to its fitness. Then evolution mechanisms like crossover and mutation are performed on the parents to create new members to the population. If a prespecified fitness value is reached, or if the fitness cannot be improved any more the algorithm is terminated; otherwise it is repeated over and over by creating new generations.

4.2 Application of GA to the Strong Stabilization Problem with an $\mathcal{H}^\infty$ Performance Constraint

In Section 2 is shown that the problem of strong stabilization of $\mathbf{P}$ with $\mathbf{K}$ which will also satisfy the $\mathcal{H}^\infty$ performance constraint has been reduced to the strong stabilization problem of $\mathbf{C}$ with $\mathbf{Q}$ in which $\mathbf{Q}$ will have to satisfy an $\mathcal{H}^\infty$ norm bound, see Figure 3. For this purpose

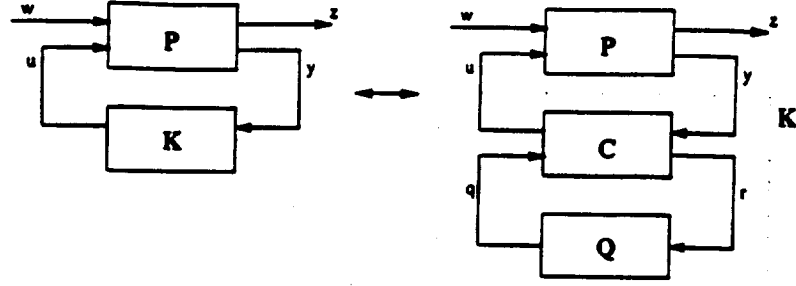


Figure 3: *Parametrization of the controller*

GA is implemented.

1. Initialize the population; randomly pick $Q = (A_q, B_q, C_q, D_q)$'s.
2. Make sure that all the Q 's result in stable A_k matrices

Until achieved, create new Q 's by using the current member (Q leading to the unstable A_k) and fittest element in the population as parents.

3. Assign fitnesses to all Q 's (as a function of $\|Q\|_\infty$, and $\lambda_{\max}\{A_k(Q)\}$)

If for any Q , $\|Q\|_\infty < \gamma$, and $\lambda_{\max}\{A_k(Q)\} < 0$ is achieved, then TERMINATE

4. Compute the new parents which will create the next generation (roulette wheel selection)
5. Construct the new generation using evolution mechanisms (crossover and mutation)

Propagate the fittest element directly to the new generation (elitism)

6. Go to Step 2.

A search for a constant Q was tried first, i.e. search for D_q only. For small sized Q 's, and for the tracking problem, this approach was observed to find a D_q matrix, which solves the problem. Since we were dealing with a constant Q , we did not need to worry about the stability, but only the norm constraint in this approach.

A search for a dynamic compensator has also been coded, and tried for the same plant. Yet, ensuring that A_k is stable with genetic operators is very time consuming. As inclusion of

a dynamic compensator significantly enlarges the parameter space, the search becomes much more time consuming (yet, it is believed that now the Q 's with the desired properties are denser in this space, and thus can be easier to find).

Recall from the previous section that the strong stabilization problem without the $\mathcal{H}^\infty$ performance constraint can also be formulated as an interpolation problem with an outer function. This approach may be utilized while initializing the population, or during the effort of creating a stable compensator which makes A_K stable, this is Task 1 defined in the previous section. A possible approach to accomplish Task 2 is to use the GA search mechanism outlined above.

5 Conclusions and Further Research Directions

Currently there is no single computationally feasible algorithm to find “best” stable $\mathcal{H}^\infty$ controller. In this project the algorithms proposed above are identified as promising methods to find such controllers. However, further research has to be performed in order to evaluate the proposed algorithms, and make them numerically efficient. The following specific tasks are proposed for further study.

Task 1. Interpolation with outer functions:

Write a Matlab based program for parameterizing all outer functions S_o such that $S_o(z_i) = 1/m_d(z_i)$, $i = 1, \dots, k$ and $\|S_o\|_\infty < \rho$ for some fixed large ρ .

Task 2. Genetic algorithms for stable $\mathcal{H}^\infty$ controller search:

Incorporate the parameterization implemented in Task 1 into the genetic algorithm which is used for finding a stable Q with $\|Q\|_\infty < \gamma$, which stabilizes G_C . For an alternative solution, write a separate GA code for searching a feasible outer S_o satisfying

$$\|C_o^{-1}(S_o^{-1} - m_d)\|_\infty < \gamma.$$

Task 3. BMI optimization methods:

In Section 2 it was demonstrated that the problem of finding a stable $\mathcal{H}^\infty$ controller can be formulated as a BMI optimization problem. But the BMI optimization problem is shown to be NP-hard. Nevertheless, for the special form of the BMIs appearing in stable $\mathcal{H}^\infty$ controller design there may be an efficient solution. We also propose to study the structure of BMIs for this problem, in particular the effect of different realizations of the central controller will be investigated.

References

- [1] Vidyasagar, M., *Control System Synthesis: A Factorization Approach*, MIT Press, Cambridge MA, 1987.
- [2] Youla, D. C., J. J. Bongiorno, Jr., and C. N. Lu, "Single-loop feedback stabilization of linear multivariable plants," *Automatica*, 10 (1974), pp. 159–173.
- [3] Doyle, J., B. Francis and A. Tannenbaum, *Feedback Control Theory*, Macmillan, New York, 1992.
- [4] Wang, Y. W., and D. S. Bernstein, " H_2 -Suboptimal stable stabilization," *Proc. of the 32nd IEEE Conf. on Decision and Control*, San Antonio TX, December 1993, pp. 1828–1829.
- [5] Toker, O., and H. Özbay, "On the structure of $\mathcal{H}^\infty$ controllers," to appear in *Proc. of the 33rd IEEE Conference on Decision and Control*, Lake Buena Vista FL, December 1994.
- [6] Doyle, J., K. Glover, P. P. Khargonekar, and B. Francis, "State space solutions to standard H^2 and H^∞ control problems," *IEEE Trans. Automatic Control*, AC-34 (1989), pp. 831–847.
- [7] Glover, K., and J. C. Doyle, "State-space formulae for all stabilizing controllers that satisfy and H_∞ -norm bound and relations to risk sensitivity," *Systems and Control Letters*, 11 (1988), pp. 167–172.
- [8] Green, M., K. Glover, D. Limebeer, and J. C. Doyle, "A J-spectral factorization approach to $\mathcal{H}^\infty$ control," *SIAM J. Control and Optimization*, 28 (1990), pp. 1350–1371.
- [9] Özbay, H., and S. Garg, "Stable $\mathcal{H}^\infty$ controller design for longitudinal dynamics of an aircraft," NASA Technical Memorandum, 106847, February 1995.
- [10] Ganesh, C., and J. B. Pearson, "Design of optimal control systems with stable feedback," *Proc. of the American Control Conference*, 1986, pp. 1969–1973.
- [11] Sideris, A., and M. G. Safonov, "Infinity-norm optimization with a stable controller," *Proc. of the American Control Conference*, 1985, pp. 804–805.

- [12] Nakayama, T., H. Ohmori, A. Sano and H. Ito, "A design of H_∞ stable controller," *Proc. of 3rd European Control Conference*, 1995, pp. 1830–1833.
- [13] Safonov, M. G., K. C. Goh, and J. H. Ly, "Control system synthesis via bilinear matrix inequalities," *Proc. of the American Control Conference*, 1994, pp. 45–49.
- [14] Grigoriadis, K., and R. E. Skelton, "Low-order control design for LMI problems using alternating projection methods," *Proc. of the 31st Allerton Conference on Communication, Control and Computing*, 1993, pp. 525–534.
- [15] Toker, O., and H. Özbay, "On the NP-hardness of solving bilinear metrix inequalities and simultaneous stabilization with static output feedback," *Proc. of the American Control Conference*, 1995, pp. 2525–2526.